

Matti Jaatinen

NORELCON METALLIOSASTON TUOTANNONSUUNNITTELU- JA OHJAUSOHJELMISTO

Opinnäytetyö
Tietotekniikka


Toukokuu 2011




MIKKELIN AMMATTIKORKEAKOULU

Mikkeli University of Applied Sciences

KUVAILULEHTI

		Opinnäytetyön päivämäärä 13.5.2011
Tekijä(t) Jaatinen Matti Juhani		Koulutusohjelma ja suuntautuminen Tietotekniikan koulutusohjelma
Nimeke Norelcon metalliosaston tuotannonsuunnittelu- ja ohjausohjelmisto		
Tiivistelmä <p>Norelco on suomalainen perheyrittys, joka suunnittelee ja valmistaa sähkönjakelukeskuksia. Tämän opinnäytetyön pohjana on Norelcolle toteuttamani ohjelmointityö. Lähtökohtana oli toteuttaa Norelcolle ohjelmisto metalliosaston tuotannonohjaukseen sekä -suunnitteluun. Ohjelmistolla on mahdollista seurata, suunnitella ja hallita Norelcon metalliosaston tuotantoa. Työn toteutukseen käytettävä teoria hankittiin ohjelmointia, käyttöliittymää ja tuotannonohjausta käsittelevistä teoksista.</p> <p>Ohjelmiston käyttöliittymän suunnittelussa pääpainona oli tehokkuus ja yksikertaisuus. Tämä toteutettiin käyttäjäläheisen testauksen avulla. Toteutuksen onnistumisen arviointi, muodostui käyttäjiltä saadusta suullisesta palautteesta. Palautteen perusteella käyttöliittymän ulkoasua muokattiin, käyttäjien toiveiden mukaisesti.</p> <p>Ohjelmiston käyttöliittymä sekä käyttäjiltä piilotettu näkymätön toiminta, toteutettiin käyttäen Delphi-ohjelmointiympäristöä. Informaation hakemiseen sekä muokkaukseen käytettiin SQL-kielellä. Delphi-ohjelmointiin käytetään Borlandin Delphi 5-kehitysympäristöä sekä SQL-tietokantana toimii IBM Informix tietokantaohjelmisto.</p> <p>Työ onnistui hyvin ja ohjelmisto otettiin käyttöön aikataulussa. Järjestelmän integrointi Norelcon jo olemassa oleviin järjestelmiin sujui melko vaivattomasti, kiitos jo olemassa olevan dokumentaation. Käytännön palaute on ollut palkitsevaa sekä antaa pohjaa uusille projekteille.</p>		
Asiasanat (avainsanat) Delphi, ohjelmointi, SQL, tuotannonohjaus		
Sivumäärä 47 + liitteet 3 sivua	Kieli Suomi	URN URN:NBN:fi:amk-201105168198
Huomautus (huomautukset liitteistä)		
Ohjaavan opettajan nimi Mynttinen Timo		Opinnäytetyön toimeksiantaja Norelco Oy

DESCRIPTION

 MIKKELIN AMMATTIKORKEAKOULU Mikkeli University of Applied Sciences	Date of the bachelor's thesis 13. May 2011	
Author(s) Jaatinen Matti Juhani	Degree programme and option Information Technology	
Name of the bachelor's thesis Norelco's production design and management program		
Abstract <p>Norelco is Finnish family-owned company that designs and manufactures power distribution systems. The purpose of this Bachelor's thesis was to produce a program for Norelco to manage, design and monitor their metal department's production. The theory needed for implementation of the work was gathered from literature based on programming, user interfaces and production management.</p> <p>The key element for designing the user interface was that it should be efficient and simple. This was accomplished by using user feedback. The success of this implementation was determined by spoken feedback received from users.</p> <p>The user interface and processes hidden from users was implemented by using the delphi development environment. Information queries and modifications was made by using the SQL language. The Delphi development environment used, was Borland Delphi 5 and the SQL database used was IBM's Informix server.</p> <p>The overall work went well and the program is currently in use. The integration of the program with the Norelco's existing system went quite smoothly thanks to the existing documentation. The hands-on feedback has been rewarding and gives base for new projects.</p>		
Subject headings, (keywords) Delphi, programming, SQL, production management		
Pages 47 + appendices 3 pages	Language Finnish	URN URN:NBN:fi:amk-201105168198
Remarks, notes on appendices 		
Tutor Mynttinen Timo	Bachelor's thesis assigned by Norelco Oy	

SISÄLTÖ

LYHENTEET

1 JOHDANTO.....	1
2 RELAATIOTIETOKANTA.....	2
2.1 Rakenne.....	3
2.2 Taulujen käsittely.....	5
2.3 Tapahtumat.....	5
2.4 Relaatiotietokannan normalisointi.....	6
3 SQL-KIELI.....	8
3.1 Taulujen määrittäminen.....	8
3.2 Hakeminen taulusta.....	9
3.3 Taulun muokkaus.....	12
3.4 Tapahtumien hallinta.....	13
4 KÄYTTÖLIITTYMÄ.....	13
4.1 Typografia.....	14
4.2 Värit.....	15
4.3 Käyttöliittymän suunnittelu.....	16
5 DELPHI 5.....	17
5.1 Olio-ohjelmointi.....	17
5.2 Object Pascal	18
5.3 Delphi 5 -kehitysalustan käyttöliittymä.....	19
5.4 Tietokantaliityntä.....	21
5.5 Käyttöliittymän tietokantakomponentit.....	22
6 TUOTANNONOHJAUS JA -SUUNNITTELU.....	23
6.1 Tuotannonohjaus.....	23
6.1.1 Tuotantolajit.....	25
6.1.2 Tuotannon tavoitteet.....	25
6.1.3 Valmistusjärjestelmä.....	26
6.2 Tuotannonsuunnittelu.....	27
7 CASE: NORELCO OY.....	28
7.1 Norelcon metalliosasto.....	29
7.2 Metalliosaston tarpeiden kartoittaminen.....	30
8 NMETAL-OHJELMISTO.....	32

8.1 NMetal-ohjelmiston rakenne.....	32
8.1.1 Metalliosien luominen.....	34
8.1.2 Tuotannon käsittely.....	37
8.1.3 Raportointi.....	44
9 PÄÄTÄNTÖ.....	46
LÄHTEET.....	47
LIITTEET	
1 Delphi 5- kehitysalustan tuottama koodi.	
2 NMetal-ohjelmiston tilausten seurantaan käytettävä raportti.	
3 NMetal-ohjelmiston tuotannon seurantaan käytettävä raportti.	

LYHENTEET

ANSI	American National Standards Institute
Buukkaus	Tilauksen kirjaaminen sekä siirtäminen tuotantoon.
NMetal	Norelcon metalliosaston tuotannonsuunnittelu- ja ohjausohjelmisto
IDE	Integrated Development Enviroment
POL-koodi	Pisteosaluettelokoodi
SQL	Structured Query Language
RAD	Rapid Application Development
RDBMS	Relational Database Management System
VCL	Visual Compoment Library

1 JOHDANTO

Ohjelmointi sekä erilaisten järjestelmien toteuttaminen on kiehtonut minua aina. Tämä käsittää järjestelmien toteuttamisen ohjelmallisesti sekä rakenteellisesti. Tästä saadun tuloksen onnistunut toteutuminen toimintaympäristössään on mielenkiintoista. Mielestäni palkitsee todeta toteuttamansa järjestelmän toiminta sekä seurata tämän toimintaa. Tämän työn toteutuksessa etenkin henkilöiden kohtaaminen sekä heidän omia tarpeitaan vastaavan ohjelmiston tuottaminen on ollut erinomaisen palkitsevaa.

Norelco on suomalainen perheyrittäjä, joka suunnittelee ja valmistaa sähkönjakelukeskuksia. Asiakaskunta muodostuu teollisuudesta, energiasektorista, liike- ja toimitiloista sekä asumisesta. Tuotteiden arvostus luodaan laajalla palvelulla, tinkimättömällä laadulla sekä jatkuvalla kehitystyöllä. Yritys sai alkunsa vuonna 1962 kun E.O. Härmäläinen aloitti elektroniikka- ja sähköalan liiketoiminnan Savonlinnassa. Nykyisin Norelcolla on toimipaikat Suomessa Vantaalla, Kuopiossa ja Savonlinnassa sekä henkilöstöä heillä on 160.

Norelco pyrkii mahdollisimman pitkälle vietyyn omavaraisuuteen tuotteiden tuotannossa. Tämä tarkoittaa Norelcon tuotteissaan käyttämien metalliosien omavaraista valmistusta raaka-aineesta alkaen. Tämä tapahtuu Norelcon metalliosastolla jolla valmistetaan suurin osa Norelcon käyttämistä metalliosista.

Monilla yrityksillä on tarpeita omiin yksilökohtaisiin ohjelmistoihin. Norelcolle tekemäni työn tarkoituksena on laajentaa Norelcon jo olemassa olevaa ohjelmistoympäristöä uudella NMetal-ohjelmistoratkaisulla (Norelcon Metalliosaston Tuotannonsuunnittelu- ja ohjausohjelmisto). Tämän työn tarkoituksena on luoda Norelcon metalliosastolle uusi ATK-järjestelmä, joka integroidaan Norelcon jo olemassa oleviin toimisto-ohjelmiin. Tällöin on mahdollista toteuttaa saumaton tuotannon eteneminen, keskustusten markkinoinnista ja tilauksesta, metalliosien tuotantoon saakka. Ohjelman työstäminen aloitettiin kesällä 2010 ja on jatkunut talven 2010-2011 aikana. Työn ohjelmisto toteutettiin käyttäen Delphi-kehitysympäristön sekä SQL-kielen yhdistelmää.

Ennen tämän työn toteutusta minulle oli ohjelmoinnista kertynyt kokemusta etenkin C-kielestä sekä hieman SQL-kielestä. Teorian soveltaminen on ollut keskeisessä ase-

massa työn toteutuksessa. Työhön käytetty teoria on koottu alan kirjoista jotka käsittelevät tämän työn aihealueita. Merkittävimpänä tiedonlähteenä olen käyttänyt Borland Delphi 5 Trainer Pro kirjaa sekä internetistä löytyviä Delphi-aiheisia keskustelupalstoja. Oikeilla työkaluilla Delphin käyttämä Object Pascal ei ole vaikea kieli opiskella, jo muutaman intensiivisen viikon opiskelun sekä käytön jälkeen, voidaan Delphillä luoda hyvinkin asiakaskohtaisia sekä tehokkaita ohjelmistoja.

Opinnäytetyön tarkoituksena on luoda kuvaus Norelcolle tekemästani NMetal-ohjelmistosta. Valmistuneen ohjelmiston avulla on tarkoitus edesauttaa Norelcon metalliosaston toimintaa nopeaksi sekä luotettavaksi. Toteutin opinnäytetyön selvittämällä SQL-kielen sekä Delphin käyttämän Object Pascal-kielen rakennetta. Käyn myös läpi Delphi-kehittimen ominaisuuksia, käsittelen käyttäjien huomioimista käyttöliittymän suunnittelussa sekä käydään läpi teoriaa tuotannonohjauksesta. Toteuttaessa työtä sain ohjelmiston tuotannonohjauksen toteuttamiseen huomattavaa opastusta työpaikanohjaajalta Ari Hämäläiseltä. Lopuksi esittelen NMetal-ohjelmiston lähtökohtia, tarpeita sekä ohjelmiston toteutusta.

NMetal-ohjelmiston kehitys tapahtui omassa kehitysympäristössä johon on varattu oma ohjelmiston kehitykseen tarkoitettu tietokanta. Tällä tavalla voidaan turvallisesti kehittää ohjelmistoa ja käyttää tuotantoa yhtäaikaaisesti, ilman että kehitystyö vaikuttaisi tuotantoon.

Kaikki NMetal-ohjelmiston liittyvät esimerkit ja käytetyt kuvat on otettu kehityskäyttöön tarkoitettusta tietokannasta. Näillä ei ole mitään yhteyttä Norelcon tuotannon, suunnittelun tai asiakkaiden kanssa. Näiden on tarkoitus antaa esimerkkiä NMetal-ohjelmiston ominaisuuksista.

2 RELAATIOTIETOKANTA

Relaatiotietokanta on tietovarasto, jonka tarkoituksena on varastoida tietoa. Tiedon järjestäminen tietovarastoon tapahtuu joukko-opillisesti. [1, s. 5 – 6.] Informaation muokkaukseen ja hakemiseen voidaan käyttää erilaisia kyselykieliä. Yleisin näistä on SQL-kieli, josta myöhemmin kappaleessa 3. [2, s. 1.]

Tässä kappaleessa käsittelen relaatiotietokantojen rakennetta sekä tarkastellaan muutamia relaatiotietokantojen huomionarvoisia yksityiskohtia. Tutkin myös relaatiotietokantojen tyyliä varastoida informaatiota sekä mitä erityishuomioita liittyy informaation käsittelyyn.

2.1 Rakenne

Relaatiotietokannat perustuvat E.F. Coddin julkaisemaan relaatiotietomalliin. Tämä malli määrittelee tietokannan teoreettisen pohjan joka rakentuu matematiikkaan, joukko-oppiin sekä predikaattilogiikkaan. Predikaattilogiikalla pystytään ilmaisemaan vain lauseita jotka ovat joko tosia tai epätosia. Relaatiotietokanta ei ole riippuvainen tietokannan fyysisestä toteutuksesta, vaan informaatio pyritään esittämään mahdollisimman loogisesti. Relaatioalgebran avulla pyritään tieto normalisoimaan, eli tieto pyritään pilkkomaan mahdollisimman loogisiin osiin. Tähän tarkoitukseen käytetään tauluja, joiden sarakkeisiin tallennetaan informaatio riveinä. Sarakkeita kutsutaan yleensä myös nimellä kenttä sekä rivejä nimellä tietue. [3, s. 6.]

Relaatiotietokannassa kaiken informaation tulee esiintyä tauluissa. Tietokannan taulut eivät ole hierarkkisessa järjestyksessä, eivätkä ne tarvitse välittää toisistaan. Taulun jokaisen rivin ja sarakkeen risteymä voidaan yksilöidä. Tätä varten tarvitsee tietää taulun nimi, mitä saraketta halutaan tarkastella sekä millä perusavaimella halutaan hakea informaatiota. Perusavaimella(primary key) tarkoitetaan sitä saraketta jonka riveillä on sellainen arvo, jota sarakkeen muilla riveillä ei ole. Avain voi koostua myös useammasta sarakkeesta. Jotta voidaan varmistua sarakkeen yksilöllisyydestä, käytetään avaimena yleensä juoksevaa numeroa. On myös mahdollista määrittää avaimena käytettävälle sarakkeelle ehtoja, jotka varmistavat sarakkeen yksilöllisyyden. [3, s. 2 – 7.]

Kuvassa 1 on esitetty yksinkertainen tietokanta jossa on taulut *nimi*, *tilaus* ja *tuote*. Kuvan taulussa *nimi* on kolme saraketta sekä neljä riviä. Taulun *nimi* avain voi olla *ni_id*, koska se eroaa muista rivin arvoista. Tälle sarakkeelle voidaan myös antaa sel-

laisia arvoja joka erottuvat muista arvoista. Perusavaimena toimiva sarake tulee aina valita siten, että sen rivien arvo on varmasti yksilöllinen kaikissa tilanteissa [2, s. 3].

nimi		
<u>ni_id</u>	ni_enimi	ni_snimi
4	Albert	Einstein
2	Blaise	Pascal
3	Albert	Edenfelt
5	Minna	Canth

tuote		
<u>tu_id</u>	tu_nimike	tu_hinta
1	5	16
2	4	20

tilaus			
<u>ti_id</u>	ti_tilaaja	ti_tuote	ti_maara
1	4	2	5
2	5	1	2

KUVA 1. Esimerkki taulut

Kuvan 1 taulussa *nimi* on sarake *ni_id*, jonka rivien arvot ovat yksilölliset. Sarakkeen *ni_enimi* ja *ni_snimi* arvot eivät välttämättä ole uniikkeja. Myöskään taulun *nimi* sarakkeiden *ni_enimi* ja *ni_snimi* yhdistelmä ei voi taata varmaa yksilöllisyyttä. Koska taulussa on sarake joka on uniikki, voidaan taulusta hakea haluttu tieto, kunhan tiedetään taulun nimi ja perusavain.

2.2 Taulujen käsittely

Tietokannassa olevaa informaatiota on myös pystyttävä käsittelemään, hakemaan sekä ylläpitämään. Relaatiotietokannoissa tämä perustuu joukko-oppiin. Relaatiotietokannat tarjoavat tähän korkean tason kielen kuten SQL. Korkean tason kielellä tarkoitetaan sitä, että kielen rakenne on kovin samankaltainen esimerkiksi englannin kielen kanssa. Relaatiotietokannassa kaikkia taulun sarakkeita voidaan käyttää hakuperusteena. [1, s. 4 – 7.]

Koskaan ei voida olla varmoja minkälaista informaatio on ja onko informaatio täydellistä. Tällöin on pystyttävä erottamaan tyhjät, NULL-arvot ja nollat. NULL ei tarkoita nollaa tai tyhjää. NULL-arvon tarkoitus on ilmoittaa, että informaatio on tuntematon. Eräs NULL-arvon huomionarvoinen olemus on, että ehto $NULL = NULL$ tuottaa aina tuloksen epätosi. NULL-arvon testaaminen tulee siis suorittaa ehdolla *IS NULL*. [2,s.15; 133.]

Informaation turvaaminen nykyisin on tärkeämpää yhä kasvavassa määrin. Relaatiotietokannat tarjoavat informaation turvaamiseen useita eri keinoja. Useimmin käytetty ominaisuus on käyttöoikeudet. Käyttöoikeuksilla voidaan kontrolloida taulujen ja sarakkeiden muokkausta sekä näistä hakua. [2, s. 10 – 15.]

2.3 Tapahtumat

Suunniteltaessa ohjelmistoja on tarkoituksenmukaista varmistua siitä, että tietokantaan syötetty tieto myös tallennetaan tietokantaan, sellaisena kuin suunnittelija haluaa sen sinne tallentuvan. Tämä on relaatiotietokannoissa toteutettu siten, että SQL-komennon suorituksen yhteydessä tarkastetaan suorituksen onnistuminen. Jos suoritus ei onnistu tai siinä on virheitä keskeytetään suoritus. Tällöin puhutaan kaikki tai ei mitään logiikasta. Relaatiotietokannoissa tällaista toimenpidettä kutsutaan nimellä tapahtumien hallinta(Transaction Management). Tämä tarjoaa menetelmän turvallisen informaation muokkaukseen ja virhetilanteiden hallittuun käsittelyyn. [2, s. 16.]

Tapahtumien hallinta tarkoittaa toimenpidettä jossa tiedon käsittelyyn osallistuvat SQL-lauseet toteutetaan niin, että nämä muodostavat yhden kokonaisuuden. Tämän kaltaista kokonaisuutta kutsutaan nimellä tapahtuma (transaction) [3. s. 16]. Tällä voidaan turvata lauseiden oikeanlainen suoritus. Tapahtumia tarvitaan käyttäjien samanaikaisuudesta(concurrency) johtuvan törmäämisen estämiseen sekä tietokannan tilan palauttamiseen edelliseen tilaan, ennen tapahtuman alkamista. [2, s. 326.]

Esimerkkinä ongelmasta joka johtuu käyttäjien samanaikaisuudesta on klassinen "lost-update problem": Olet ostamassa junalippua automaatilta Sodankylään. Automaatti tarjoaa viimeistä paikkaa junassa. Toisaalla samaan aikaan myös toinen henkilö on ostamassa lippua Sodankylään ja myös hänelle esitetään viimeinen paikka. Ostat lipun ja järjestelmä päivittää lipun myydyksi, mutta hetkeä myöhemmin toisaalla henkilö ostaa myös saman lipun. Tällöin henkilön tekemä ostos ylikirjoittaa sinun ostoksesi ja virhe on syntynyt. Tämä oli esimerkki samanaikaisuuden aiheuttamasta ongelmasta, joka relaatiotietokantojen on kyettävä hallitsemaan. [2, s. 327.]

SQL-ohjelmoija voi hyödyntää tapahtumia työssään myös konkreettisemmin. Hän voi toteuttaa tapahtumien avulla kokonaisuuksia, jotka muodostuvat useammasta SQL-lauseesta. Tämänkaltaisen toimenpide voisi olla varastosta komponenttien siirtäminen projektiin. Tapahtuma siis muodostuu kahdesta toimenpiteestä, vähennetään varaston komponenttien määrää ja lisätään projektin komponenttien määrää. Tässä tilanteessa on mahdollista suorituksen keskeytyminen sillä hetkellä, kun varaston komponentti määrää on muokattu. Koska vain varaston komponenttimäärää on muokattu; on syntynyt virhe. Tämä voidaan estää aloittamalla tapahtuma ennen varaston saldon muokkausta ja lopettamalla tapahtuma, projektin komponenttien määrän päivityksen jälkeen. Virhetilanteessa komponenttien määriä ei tallenneta, koska tapahtuma keskeytetään. Mikäli kummatkin lauseet suoritetaan onnistuneesti tallennetaan myös samalla muutokset tietokantaan. [2, s. 328.]

2.4 Relatiotietokannan normalisointi

Koska relaatiotietokannoissa informaatio esitetään tauluissa, on olemassa menetelmä jolla taulut voidaan muodostaa niin, että näissä ei esiinny päällekkäisyyttä. Tätä nimi-

tetään tietokannan normalisoinniksi. Normalisoinnilla tarkoitetaan yleensä menettelyä jossa taulut jaetaan tilanteesta riippuen, joko yhteen tai useampaan tauluun. Tämän tarkoituksena on vähentää informaation päällekkäisyyttä (data redundancy). Normaalitasot toimivat niin, että ylempi taso sisältää alemman tason ominaisuudet. Normaalitasoja on useita, mutta yleensä nämä jaetaan viiteen eri tasoon. Yleensä suunnittelijat kuitenkin päätyvät käyttämään vain neljää ensimmäistä tasoa. [2, s.39 – 40]. Tässä käydään läpi nämä normalisoinnin neljä ensimmäistä tasoa.

Saavuttaakseen ensimmäisen normaalitason, taulun rivien ja sarakkeiden leikkauskohdan täytyy muodostaa vain yksi arvo, joka on atominen. Atomisella tarkoitetaan arvoa joka muodostuu sen kokonaisuuden osasta. Esimerkiksi nimi voidaan jakaa kahteen atomiseen osaan, etu- ja sukunimi. Jotta rivien ja sarakkeiden leikkauskohta olisi yksilöllinen, taulussa ei saa olla toistuvia rivejä. Toinen normaalitaso koskee vain tauluja joissa perusavain muodostuu kahdesta tai useammasta sarakkeesta. Normaalitaso vaatii, että taulun jokainen sarake joka ei kuulu perusavaimen, tulee olla riippuvainen koko perusavaimesta. [2, s. 40 – 42.]

Kolmannessa normaalitasossa taulujen sarakkeet jotka eivät ole avaimia, eivät saa riippua taulun toisista sarakkeista, jotka eivät ole avaimia. Tällöin taulun sarakkeet jotka eivät ole avaimia, täytyy riippua vain avaimesta. Neljännessä normaalitasossa taulu saa muodostua vain siten, että taulun ei avain sarakkeet voidaan rinnastaa taulun muihin sarakkeisiin. [2, s. 43 – 45.]

Tutkimalla luvussa 2.1 olevan kuvan 1 tauluja, voidaan huomata näiden olevan ainakin normaalitasossa 1. Tämä voidaan todeta sillä kaikkien taulujen sarakkeet muodostavat atomisen arvon. Taulut ovat myös toisessa normaalitasossa, koska taulujen perusavain muodostuu vain yhdestä sarakkeesta. Taulut ovat myös kolmannessa normaalitasossa, koska kaikki taulujen sarakkeet riippuvat vain taulujen perusavaimista. Tutkimalla tauluja huomataan myös että taulut ovat neljännessä normaalitasossa, sillä taulujen ei avain sarakkeet ovat rinnastettavissa taulujen muihin sarakkeisiin.

Koska käytännössä aina ei voida suunnitella toteutusta siten, että tietokanta olisi normalisoidussa muodossa. Tämä voi johtua esimerkiksi suorituskyvystä tai sellaisten taulujen välttämisestä joihin normalisoidussa muodossa kohdistuisi paljon

toimintaa(hakuja tai muutoksia). Tätä menetelmää kutsutaan relaatiomallin denormalisoinniksi. [3, s. 7.]

3 SQL-KIELI

Suurimmassa osassa markkinoilla olevista relaatiotietokantatuotteista käytetään kyselykielenä SQL-kieltä. Jotta valmistajat erottuisivat joukosta, he lisäävät tuotteisiinsa omia SQL-laajennuksiaan. Näillä laajennuksilla voidaan saavuttaa tietyissä tilanteissa parempi suorituskyky kilpailijoiden tuotteisiin nähden. Suurimmassa osassa relaatiotietokantaohjelmistoja keskeisimmät komennot ja näiden toiminta on silti saatettu yhteensopiviksi SQL-92n kanssa. SQL-92 on ANSI(American National Standards Institute) julkaisema standardi SQL-kielestä. ANSI julkaisi standardin 1992 ja se on tällä hetkellä pisimmälle levinnyt sekä yhteensopivin SQL-kielen standardi. [1, s. 2; 2, s. xxvii.]

Yleensä SQL-kielessä varatut sanat eivät ole riippuvaisia kirjainten koosta. On kuitenkin selvyuden vuoksi hyvä kirjoittaa SQL-kieltä koskevat sanat isolla ja muut, informaatiota sekä tauluja koskevat sanat pienellä. Tällöin on helppo todeta mitkä sanat tulisi kuulua SQL-kutsuun sekä kutsun parametreihin. [1, s. 14.]

3.1 Taulujen määrittäminen

Jotta tauluun voidaan siirtää informaatiota tulee tietää minkälaista informaatiota varastoidaan. Taulujen rivien datan tyyppi määritetään taulun sarakkeissa. Seuraavalla listalla on esitetty tyypillisiä SQL-tietokannan tietotyyppejä [3, s. 3.]:

- Float ja integer ovat tietotyyppejä jotka varastoivat numerotietoa. Näiden valintaan vaikuttaa haluttavan informaation tarkkuus. Esimerkiksi float kykenee esittämään desimaalipilkun mutta integer ei.
- Char- tai varchar -tietotyyppisiin varastoidaan merkkitietoa. Näistä char voi vain varastoida yhden merkin ja varchar voi varastoida useamman.
- Blob-tietotyyppillä voidaan varastoida binääristä informaatiota tai suuria määriä määrittelemätöntä informaatiota.

- Date- ja datetime -tyypeillä voidaan varastoida päivämäärään ja aikaan liittyvää informaatiota. DateTime voi varastoida päivämäärän ja kellonajan sekä date voi varastoida päivämäärän.

SQL-92 tarjoaa taulujen määrittämiseen kolme keskeisintä komentoa, CREATE, ALTER ja DROP. Näistä komennosta CREATE- komento luo taulun. ALTER- komento muuttaa taulun rakennetta sekä DROP- komento poistaa taulun ja kaikki poistettavassa taulussa olevan datan. [1, s. 82 – 89.]

Kappaleessa 2.1 esitetyssä kuvassa 1, oleva taulu *tuote* voidaan luoda kuvassa 2 esitetyllä CREATE-komennolla. Tässä komennossa sarakkeesta *tu_id* tehdään perusavain määrittämällä tälle optio *PRIMARY KEY*. Monet relaatiotietokantatuotteet asettavat automaattisesti *UNIQUE INDEX*- option perusavaimille [1, s. 112]. Taulun sarakkeen *tu_nimike*, tyyppi on *CHAR* sekä tämä voi varastoida 20 merkkiä tietoa. Varmistuaan että tuotteelle annetaan nimi, asetetaan nimelle optio *NOT NULL*.

```
CREATE TABLE tuote (tu_id INTEGER PRIMARY KEY,
                    tu_nimike CHAR(20) NOT NULL,
                    tu_hinta INTEGER)
```

KUVA 2. CREATE- komennolla luotu taulu tuote

3.2 Hakeminen taulusta

SQL-kielen keskeisin käsky on SELECT-käsky. Tällä käskyllä on mahdollista toteuttaa sovellus, jossa ei tarvita informaation muokkaamista. Esimerkkinä tällaisesta sovelluksesta on yksinkertainen raportointi väline, joka muodostaa tietokannan datasta raportin. Käskyn perusrakenne on esitetty kuvassa 3.

```
SELECT [sarakkeet pilkulla erotettuina tai *] FROM [taulun nimi] WHERE [ehtolauseet].
```

KUVA 3. Esimerkki SELECT-lauseen rakenteesta

SELECT-käskyn aivan ehdottomia osia ovat SELECT ja FROM, muut osat ovat valinnaisia. Esimerkiksi komento: *SELECT * FROM tuote* valitsee kaikki sarakkeet sekä rivit taulusta *tuote*. Sarake valitsimeksi on valittu asteriski-merkki, joka hyväksyy kaikki FROM-osassa esitettyjen taulujen sarakkeet. SQL-lauseella on myös mahdollista rajata hakua. Tällöin käytetään WHERE-ehtolauseetta. Esimerkkinä valitaan kappaleessa 2.1 esitetyn kuvan 1 taulusta *tilaus*, ne rivit joissa ollaan tilattu tuotetta 2. Tällöin komento on muotoa: *SELECT * FROM tilaus WHERE ti_tuote = 2*. [1, s. 14 – 21.]

Aina ei ole tarpeen palauttaa kaikkia taulun sarakkeita tai voi olla tarve hakea vain tiettyjä sarakkeita. Tällöin on tärkeää pystyä valitsemaan mitä sarakkeita halutaan tulokseen, käytettäessä SQL-kieltä tämä onnistuu helposti. Esimerkiksi haluttaessa valita edellisen haun sarakkeet *ti_tuote* ja *ti_maara*, muodostetaan lause, joka on muotoa: *SELECT ti_tuote,ti_maara FROM tilaus WHERE ti_tuote = 2*. Tämä palauttaa vain sarakkeiden *ti_tuote* sekä *ti_maara* rivit, joissa on tilattuna tuotetta 2.

Relaatiotietokannassa on tärkeää pystyä liittämään tauluja helposti keskenään. Tähän tarkoitukseen on SQL-kielessä erilaisia tapoja. Eräs keskeisimmistä näistä on JOIN-lause. Komennon syntaksi on esitetty kuvassa 4.

"SELECT [sarakeet] FROM [taulu] [[liitoksen tyyppi] JOIN [liitettävä taulu] ON [liitoksen ehtolause]] WHERE [ehtolauseet]".

KUVA 4. Esimerkki liitoksen käytöstä

Liitoksen tyyppejä voivat olla CROSS JOIN, INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN ja FULL OUTER JOIN. Liitoksella CROSS JOIN toteutetaan karteesinen liitos. Karteesisella liitoksella tarkoitetaan liitosta, joka palauttaa kaikki mahdolliset rivien yhdistelmät. INNER JOIN liitos palauttaa CROSS JOIN liitoksen tuloksista ne tulokset, joiden liitoksen ehtolause on tosi. [1, s. 50 – 52.]

Esimerkkinä JOIN-komennon käytöstä voisi olla tilanne, jossa haetaan kaikki ne tuotteet, joita on tilattuna enemmän kuin 5. Tämän haun SQL-koodi olisi esimerkiksi seuraava: *SELECT * FROM tilaus A JOIN tuote B ON A.ti_tuote = B.tu_tuote WHERE A.ti_maara > 5*. Tämä palauttaisi kaikki ne taulun tilaus rivit joiden *ti_maara* on suu-

rempi kuin 5. Tämä myös liittyy näihin riveihin taulun *tuote* ne rivit, joiden *tu_tuote* vastaa taulun *tilaus* sarakkeen *ti_tuote* rivin arvoa.

Informaatiota tarkastellessa on hyvä pystyä järjestämään data tiettyyn järjestykseen. Tarkastellessa tuotantoon siirtyviä tuotteita, on hyvä pystyä järjestämään tuotteet jollain tavalla, esimerkiksi tuotekoodin tai värin perusteella. Tähän tarkoitukseen SQL tarjoaa ORDER BY -lauseen. Tällä voidaan järjestellä SQL-hausta saatu tulos tiettyjen sarakkeiden mukaan. Tällöin voitaisiin järjestellä esimerkiksi tuotteet niiden tuotekoodin mukaan, joko nousevassa tai laskevassa järjestyksessä. Oletusarvoisesti ORDER BY palauttaa sarakkeen nousevassa järjestyksessä (ASC), tämä voidaan myös asettaa laskevaan järjestykseen (DESC). [1, s. 18.]

Esimerkkinä SQL-kielen ORDER BY -lauseesta esitän tilanteen jossa kappaleessa 2.1 esitetyn kuvan 1 sisältämästä taulusta *tilaus*, haetaan kaikki *ti_tuote* sarakkeet nousevassa järjestyksessä: *SELECT ti_tuote FROM tilaus ORDER BY ti_tuote*.

Kun voidaan yhdistää useita eri tauluja, sekä voidaan muokata näiden esitys järjestykseltä. Voi tulla tarve ryhmitellä dataa niin että, hausta saatu data ryhmitellään jollakin sarakkeella. Tällöin saadaan tuloksena rivejä joissa yhdessä ryhmässä kaikilla riveillä on sama arvo ryhmitetyllä sarakkeella. Tähän käytetään GROUP BY -määrettä. Haussa jossa käytetään sarakkeeseen GROUP BY- määrettä, saadaan tulokseksi sarakkeeseen kaikki eri esiintymät kertaalleen. [1, s. 38.]

Esimerkkinä voisi olla tilanne jossa halutaan tarkastaa kappaleessa 2.1 esitetyn kuvan 1 taulusta *tilaus*, mitä eri komponentteja on tilattuna. Tällöin käytetään seuraavanlaista hakua: *SELECT * FROM tuote WHERE tu_id IN (SELECT ti_tuote FROM tilaus GROUP BY ti_tuote) ORDER BY tu_nimike*. Tämä palauttaa vain kertaalleen taulusta *tuote* ne rivit, joita esiintyy taulussa *tilaus* ja järjestää tästä muodostuvan listan sarakkeen *tu_nimike* mukaan.

3.3 Taulun muokkaus

SQL-kielessä informaation muokkaamiseen on useita komentoja. Tässä tarkastellaan näistä kahta komentoa: INSERT ja UPDATE. Näistä komennoista INSERT- komento lisää tietokannan tauluun uutta tietoa ja UPDATE muokkaa jo olemassa olevaa. INSERT- käskyn syntaksi on esitetty kuvassa 5.

```
"INSERT INTO [taulun nimi] [(sarakkeet pilkulla erotettuina)] VALUES [(tauluun vietävä informaatio pilkulla erotettuna)]"
```

KUVA 5. Esimerkki INSERT- komennon syntaksista

Esimerkkinä INSERT-käskystä on tilanne jossa halutaan tehdä uusi tilaus. Tällöin lause voisi olla seuraavankaltainen, *INSERT INTO tilaus (ti_tuote,ti_maara,ti_tila) VALUES (2,5,0)*.

Olemassa olevaa informaatiota voidaan myös muokata. Tähän tarkoitukseen SQL-kielessä on UPDATE-komento jota voidaan käyttää tilanteissa, joissa halutaan muokata jo olemassa olevaa dataa. UPDATE-komennon syntaksi on esitetty kuvassa 6.

```
"UPDATE [taulun nimi] SET [sarake1 = arvo1],[sarake2 = arvo2] WHERE [ehtolause]"
```

KUVA 6. Esimerkki UPDATE-komennon syntaksista

UPDATE-komennossa muokataan ehtolauseen määrittämän rivin sarakkeita sarake1 ja sarake2. Sarakkeisiin sijoitetaan arvot arvo1 ja arvo2. Lause voi kohdistua joko yhteen tai useampaan sarakkeeseen. Sarakkeiden sijoitukset on erotettava pilkulla. Mikäli taulun mikään rivi ei täytä UPDATE-komennon ehtolauseetta, ei UPDATE-komento tee mitään. UPDATE-komennolla voidaan myös vaikuttaa useisiin taulun riveihin. [1, s. 96]

Tauluista on myös mahdollista poistaa rivejä. Tämä tapahtuu käyttämällä DELETE-komentoa. DELETE-komennon käytössä kannattaa olla varovainen ja miettiä mikä on DELETE-komennon ehtolauseen tulos. DELETE-komennolla on mahdollista poistaa

useita rivejä yhdellä komennolla. Tästä syystä on oltava erityisen varovainen esimerkiksi ehtolauseessa käytettävän asteriski-merkin käytössä. [1, s. 99 – 100.]

3.4 Tapahtumien hallinta

Virhetilanteiden toteaminen ja hallinta on eräs SQL-kielen sekä relaatiotietokantojen keskeisimpiä ominaisuuksia. Tätä varten SQL-kielessä on COMMIT- ja ROLLBACK -komennot. COMMIT-komento päättää tapahtuman sekä ilmoittaa onnistuneesta suorituksesta. ROLLBACK-komento ilmoittaa virheellisestä suorituksesta sekä estää tapahtuman suorituksen. Relaatiotietokannoissa joissa noudatetaan SQL-92sta, tapahtuma alkaa aina edellisestä COMMIT-käskystä ja päättyy joko COMMIT- tai ROLLBACK -käskyyn. [1, s. 102.]

Tapahtumien käsittelyä voidaan kontrolloida muokkaamalla tietokannan *transaction isolation level*- asetusta. Tällä voidaan määrätä tietokannan käyttäytyminen tapahtumissa. Tälle voidaan antaa esimerkiksi arvo *dirty read*, joka mahdollistaa keskeneräisten tapahtumien lukemisen tietokannassa. *Isolation level* tulisi aina asettaa harkiten järjestelmän tarpeita vastaavaksi. [3, s. 16 – 17.]

4 KÄYTTÖLIITTYMÄ

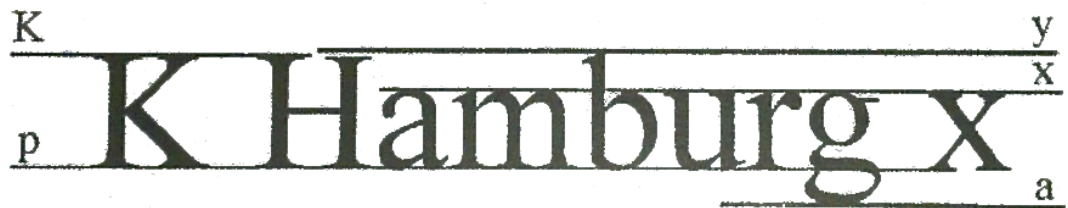
Helppokäyttöisen sekä tehokkaan käyttöliittymän suunnittelu ja toteutus on aikaa vievää sekä vaikeaa. Suunnittelijan tulee tuntee käyttäjien tarpeet sekä toimintaympäristö. Meistä jokainen on erilainen ja meillä on erilaiset kyvyt havainnoida fyysisistä että loogista ympäristöämme. Tähän vaikuttavat omat fyysiset että henkiset kykymme. Tällöin on tärkeätä ymmärtää käyttäjien ympäristö sekä heidän mahdollinen informaation tulkintatapansa. Mikäli käyttöliittymää ei pystytä tulkitsemaan, tarkoittaa se sitä että, käyttöliittymän tarjoama sisältö turhaa, koska käyttäjä ei pysty käyttämään tämän sisältämää informaatiota. Hyvin suunniteltu käyttöliittymä ei rasita käyttäjää ja on hyvin ennustettavissa. Tällöin käyttäjän ei tarvitse välittää käyttöliittymästä vaan hän voi keskittyä työhönsä. [4, s. 1; 5, s. 9 – 10.]

4.1 Typografia

Kun toteutetaan käyttöliittymää, on kirjainten oikeaa valintaa painotettava. Käytössä on hyvin suuri joukko erilaisia kirjaimia ja näistä oikean valinta käyttötarkoitukseen voi olla vaikeaa. Typografialla tarkoitetaan pääasiassa kirjainten sekä näiden käytön suunnittelua. Käyttöliittymän suunnittelijan kannalta tärkeää on osata valita oikeanlainen kirjasin. [4, s. 13.] Omien kirjainten tuottaminen on harvoin tarpeen, mutta tämäkin voidaan joutua tekemään jotain erityistä tehtävää varten, esimerkiksi viivakoodin toteutuksessa.

Yleisesti kirjainten muoto on joko päätteellinen tai päätteetön. Päätteellisiä kirjaimia (Serif) nimetään myös antikvaksi. Käyttöliittymän toteutuksessa päätteelliset voivat olla pienikokoisina vaikeaselkoisia ja tämä on huomioitava etenkin kun toteutetaan käyttöliittymiä joissa on paljon sisältöä. Päätteettömät kirjaimet (Sans Serif) on tarkoitettu alunperin julistekäyttöön ja soveltuvat myös hyvin käyttöliittymiin. [4, s. 14.]

Typografiassa käytetään mittajärjestelmää joka on käytössä painetuissa tuotteissa. Tässä järjestelmässä kirjainkoot sekä linjojen paksuudet ilmoitetaan typografisina pisteinä. Eurooppalaisessa Didot-järjestelmässä piste on 0,376 mm ja amerikkalaisessa Pica-järjestelmässä 0,352 mm. Kuvassa 7 on esitetty kirjasinten mittaukseen käytettävät kirjainkorkeuden perusmittaviivat.



KUVA 7. Kirjainkorkeuden perusmittaviivat [4, s. 16]

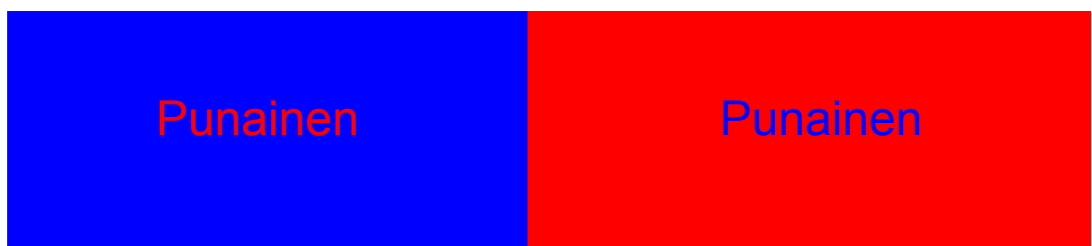
Kuvassa 7 esitetyt perusmittaviivat muodostuvat kirjainten korkeudesta K, peruslinjasta p, ylälinjasta y, keskilinjasta x ja alalinjasta a. Kirjainten kirjainkoko määritetään linjojen y sekä a erotuksena. Kirjainten riviväli on perusviivojen väli. [4, s. 16.]

4.2 Värit

Värien valinnalla on väliä etenkin käyttöliittymän suunnittelussa. Vastaan tulee välttämättä tilanteita joissa täytyy pystyä ilmoittamaan käyttäjälle jostain. Värit ovat hyviä keinoja tähän tarkoitukseen. Värejä ei kuitenkaan yksinään saa käyttää ilmoitukseen, sillä ei voida olla täysin varmoja kykeneekö käyttäjä havainnoimaan värit. Voihan olla että käyttäjä on juuri kyseiselle väriyhdistelmälle värisokea. Tätä varten käyttöliittymään on toteutettava myös menetelmä, jolla kaikkien voi olla mahdollista saada viesti. Koska ihmisistä useimmat kuitenkin erottavat värejä melko hyvin, voidaan näitä käyttää tehokkaasti ilmoitukseen. Kuten jo mainittiin, täytyy tällöin huolehtia tämän lisäksi myös tavasta jolla kaikki käyttäjät voivat huomata sen. [4, s. 34.]

Värien käytössä on myös huomioitava kulttuurilliset näkökohdat. Esimerkiksi Suomessa punainen osoittaa vaaraa tai huomiota sekä vihreä on hyväksymisen ja sallimisen väri. Toisaalta myös esimerkiksi Kiinassa valkoinen on surun väri ja punainen on onnen väri. [4, s. 36.]

On myös varottava tiettyjen väriparien valintaa. Esimerkiksi sinisen ja punaisen värien käyttöä yhtäikaa tulisi välttää. Tässä tilanteessa ihmisen silmä ei pysty muodostamaan tarkkaa kuvaa sinisestä ja punaisesta. Tämä johtuu silmän lihaksen toiminnasta. [5, s. 399 – 401.] Kuvassa 8 on esitetty havainnollistava esimerkki punaisen ja sinisen käytöstä.



KUVA 8. Esimerkki huonosta värivalinnasta

4.3 Käyttöliittymän suunnittelu

Hyvä käyttöliittymä ei hukuta käyttäjää sellaisella informaatiolla mitä käyttäjä ei tarvitse. Tällöin esitettävän informaation rajausta on tehtävä huolellisesti. On arvioitava sekä kysyttävä käyttäjiltä mitä informaatiota he tarvitsevat useimmiten käyttöliittymän käytössä. Tämä informaatio on tuotava esiin mahdollisimman selkeästi. Tämän lisäksi voi käytön aikana olla tarpeen esittää sellaista informaatiota, mitä käyttäjä ei tarvitse aina, mutta voi tarvita tätä joissain tilanteissa. Tällöin on hyvä toteuttaa käyttöliittymä siten, että hakuaskelten tai -kerrosten määrä rajoittuu 3-4 askeleeseen. Mitä vähemmän käyttäjän tarvitsee tehdä työtä informaation esille saantiin, sen parempi. [4, s. 59.]

Käyttöliittymä on rajapinta käyttäjän sekä järjestelmän välillä. Tällöin käyttöliittymän tulee olla avuksi käyttäjälle järjestelmän käytössä, mutta silti hänen kontrolloitavissaan. Eräs huomiotavimpia ominaisuuksia tässä on käyttöliittymän vasteaika. Tällä tarkoitetaan sitä missä ajassa käyttöliittymä vastaa käyttäjän tekemiin toimenpiteisiin. Tässä tulisi pyrkiä sellaiseen toteutukseen jossa vasteaika olisi mahdollisimman lyhyt, pisimmillään tämä voi kuitenkin olla sekunteja. [4, s. 59.]

Vuorovaikutus käyttöliittymän ja käyttäjän välillä tulisi olla mahdollisimman looginen sekä ennalta arvattava. Tällöin henkilö pystyy helposti omaksumaan käyttöliittymän toiminnan. Tämä lisää käyttöliittymän ja käyttäjän välisen informaation määrää, koska käyttäjän ei tarvitse olla kokoajan seuraamassa mitä käyttöliittymä tekee. [4, s. 63.]

Käyttäjien tiedon määrän hallintatarvetta voidaan tehokkaasti vähentää käyttämällä käyttöliittymässä valmiita listoja. Tällä tarkoitetaan graafisia komponentteja joiden avulla käyttäjä voi valita listalta halutun datan. Tällä voidaan myös vähentää käyttäjien tekemien virheiden määrää, sillä listasta valittava arvo on ennalta määrätty. [5, s. 265 – 266.]

5 DELPHI 5

Delphi 5 on niin sanottu RAD-työkalu (Rapid Application Development) jonka avulla on mahdollista kehittää tehokkaita ohjelmia nopeasti. Delphi on visuaalinen sovelluskehitin joka piilottaa alleen huomattavan määrän alemman tason toiminnallisuutta. Tämän ansiosta Delphillä voidaan toteuttaa hyvinkin nopeasti ohjelmia, koska ohjelmoijan ei tarvitse itse kirjoittaa koko toiminnallisuutta, lähtien alimmilta tasoilta aina ongelman ratkaisuun. [6, s. 4.]

5.1 Olio-ohjelmointi

Olio-ohjelmat ovat ohjelmia jotka muodostuvat olioista. Oliot ovat kokonaisuuksia jotka paketoivat toiminnallisuuden ja tähän liittyvän informaation. Olion proseduureja kutsutaan metodeiksi tai operaatioiksi. Proseduuri on osa oliota, joka tekee jonkin tietyn asian. Vain metodien tulee pystyä muokkaamaan olion sisäistä informaatiota. Olion tulee suorittaa toiminto vasta kun se saa asiakkaalta pyynnön. Pyyntöjen tulee olla ainoa tapa saada olio suorittamaan metodeja. Kun nämä ovat toteutettu, sanotaan olion sisäisen tilan olevan kapseloitu. Vaikein tehtävä oliopohjaisessa suunnittelussa on järjestelmän osiin pilkkominen. Tähän vaikuttavat suuresti ulkoiset riippuvuudet ja kapselointi. [7, s. 11.]

Olion rajapinta muodostuu sen kutsumallien joukoista. Rajapinnan tarkoitus on kuvata olion kaikki pyynnöt. Rajapinta voi myös sisältää muita rajapintoja. Oliolle voidaan lähettää informaatiota vain sen rajapinnan kuvaamalla tavalla. Kun asiakas lähettää oliolle informaatiota, riippuu pyynnön toteutus sen sisällöstä että vastaanottajasta. Ajonaikaista informaation yhdistämistä olioon tai sen metodiin kutsutaan dynaamiseksi sidonnaksi. Tällöin informaatiota ei sidota ennen sen käsittelyä mihinkään tiettyyn toteutukseen. Tällä voidaan toteuttaa sellaisia olioita, joilla on identtisiä rajapintoja mutta näiden toiminnot ovat erilaisia. Tällaisia menetelmiä soveltavia järjestelmiä sanotaan monimuotoisiksi. [7, s. 13 – 14.]

Olio määritellään luokan avulla. Luokka määrittää olion sisäisen datan, toiminnan sekä rajapinnan. Olio luodaan luomalla luokan kuvauksen perusteella ilmentymä. Tä-

män yhteydessä varataan oliolle tämän tarvitsemat resurssit. Luokasta voidaan luoda olioita niin paljon kuin on resursseja. Uusien luokkien luontiin voidaan myös käyttää periytymistä. Tällöin voidaan periä uuteen aliluokkaan ylikuokan ominaisuudet. Ylikuokka pysyy tällöin muuttumattomana, mutta aliluokka sisältää ylikuokan kuvauksen. On myös mahdollista lisätä aliluokkaan uusia metodeja ja tämä ei vaikuta ylikuokkaan. Mutta jos ylikuokkaan lisätään metodeja, tulevat ne automaattisesti aliluokkaan. Aliluokan on myös mahdollista korvata ylikuokan metodeja. On myös mahdollista toteuttaa abstrakteja luokkia. Nämä luokat toimivat rajapintoina aliluokille. Näistä luokista ei voi luoda ilmentymiä. [7, s. 15 – 16.]

5.2 Object Pascal

Delphin käyttämä Object Pascal on Borlandin kehittämä laajennus aikoinaan suosittua Pascal-ohjelmointikielestä. Pascal-ohjelmointikieli oli alkuperin tarkoitettu opetuskäyttöön, tämän seurauksena Pascalin ulkoasu on yksinkertainen, siisti sekä tällä on korkeantason ilmaisujen käyttömahdollisuus. Koska Pascal ei sisällä itsessään kirjastofunktioita, sisältää lähes jokainen Pascal-kielen toteutus oman kirjastolaajennuksensa, esimerkiksi tiedostojen tai merkkijonojen käsittelyyn. Suosituin Pascal-toteutus oli Borlandin Turbo Pascal. Borland jatkoi Turbo Pascalin kehitystä ja toteutti Turbo Pascalin pohjalta visuaalisen sovelluskehittimen Delphi 1.0n. [6, s. 66 – 68.]

Object Pascal on tyyppitarkka kieli. Tämä tarkoittaa että esimerkiksi integer-tyyppiä olevaa muuttujaa ei voi suoraan sijoittaa string-muuttujaan, vaan tarvitaan tähän funktio joka suorittaa muunnoksen. Tämä ei kuitenkaan estä ohjelmoinnissa tapahtuvan ylivuotoilmiön mahdollisuutta. Ylivuotoilmiöllä tarkoitetaan tilannetta jossa informaatiota on liikaa sille varattuun säilöön nähden. Esimerkiksi kuvan 9 koodi aikaansaa ylivuodon. Tämä perustuu tilanteeseen jossa kaksitavuiseen SmallInt-tyyppiä olevaan muuttujaan x , sijoitetaan ensin luku 32677 ja tämän jälkeen lisätään luku 5. Tällöin on tapahtunut ylivuoto. Estääkseen mahdollisten ylivuotojen tapahtumisen ohjelmoijan on suunniteltava koodinsa ja mietittävä mitä arvoja muuttujat voivat saada. [8, s. 27 – 30.]


```

procedure overflow();
var
  x : SmallInt;
begin
  x := 32677;
  x := x + 5;
end;

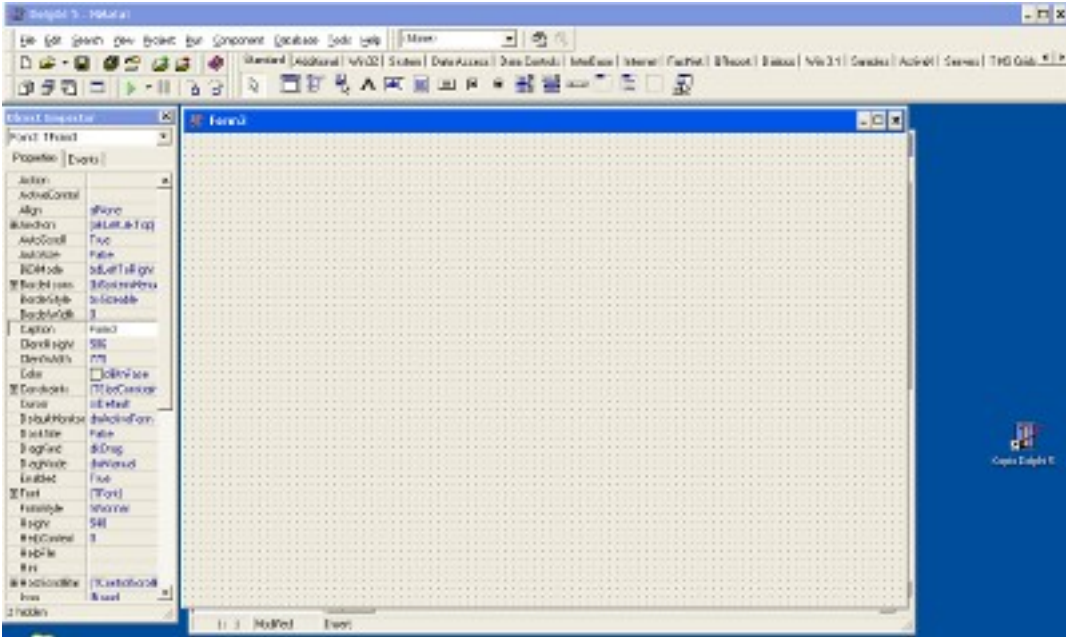
```

KUVA 9. Datat ylivuototilanne

Liitteenä 1 on esitetty Delphi 5 -kehitysalustan tuottama Object Pascal- koodi. Tässä ohjelmakoodi alkaa *unit* määrittelyllä. Tämä määrittää lähdekoodin yksikön ja tämän nimen. Delphin graafisilla sovelluksilla täytyy olla aina vähintään yksi lomake ja tällä yksikkö. Delphissä voi olla myös ei graafisia yksiköitä, jotka sisältävät vain lähdekoodia. Yksikön jälkeen tulee *interface* joka määrittää yksikön esittelyosan. Tällä osalla määritetään ne yksikön osat, joita halutaan käyttää muissa yksiköissä. Käytettävien yksiköiden määrittäminen tapahtuu *Uses*-sanalla. Tämä määrittää mitä yksiköitä yksikkö käyttää ja on näistä riippuvainen. Yksikön määrittämien tyyppien esitleminen tapahtuu lohossa *type*. Ohjelman toteutusosa alkaa *implementation*-sanalla sekä loppuu *end.*-sanaan. [8, s. 17 – 20.]

5.3 Delphi 5 -kehitysalustan käyttöliittymä

Delphin käyttöliittymä muodostuu neljästä erillisestä osasta, pääikkuna, olio-editori, työtila sekä koodi-editori. Pääikkuna sisältää työkalupainikkeet sekä komponenttipaletin. Delphin työkalurivien avulla hallitaan ohjelmistoprojektia sekä tällä voidaan myös lisätä työtilaan haluttuja komponentteja. Komponentit ovat binäärisiä ohjelmiston osia jotka suorittavat jonkin tietyn tehtävän. Esimerkiksi tekstikehys tai tietokantakomponentti. Olio-editorilla voidaan muokata komponenttien ominaisuuksia, tämän ikkunan muokkausmahdollisuudet riippuvat kokonaan komponentin toteutuksesta. Työtilassa muokataan tuotettavan ohjelman käyttöliittymää. Koodieditorilla muokataan ohjelman lähdekoodia ja tällä voidaan tehdä sellaista toiminnallisuutta jota olio-editori ei pysty tuottamaan. [8, s. 7 – 11.]



KUVA 10. Delphi 5 -käyttöliittymän oletusnäkymä

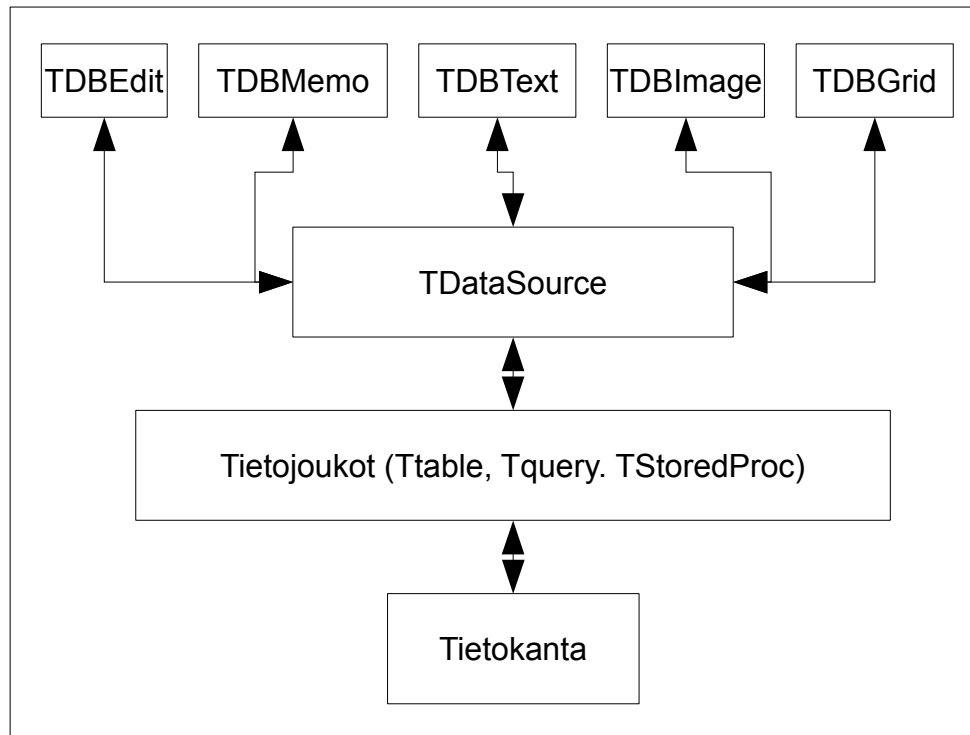
Ohjelman ajaminen tapahtuu kuvan 9 Run-pikapainikkeesta. Tällöin Delphi kääntää ja linkittää ohjelman. Delphi ei käänne muita ohjelman osia kuin ne, mitä on muokattu edellisen käynnöksen jälkeen. Tämä säästää aikaa etenkin suurissa ohjelmistoprojekteissa. Delphissä on kolme vaihtoehtoa joilla voidaan kontrolloida ohjelman kääntämis- ja linkittämisoperaatiota. Nämä ovat Syntax Check, Compile ja Build. Syntax Check tarkistaa vain ohjelmakoodin syntaksin oikeellisuuden. Tällä voidaan nopeasti tarkastuttaa ohjelmakoodissa olevat syntaksi- eli kirjoitusvirheet. Compile-vaihtoehto kääntää vain muuttuneen koodin sekä linkittää ohjelman. Tämä on tarkoitettu tilanteeseen jossa halutaan valmis ohjelma nopeasti. Build-valinta kääntää kaikki projektiin kuuluvan lähdekoodin, riippumatta onko koodi muuttunut. [8, s. 126 – 127]

5.4 Tietokantaliityntä

Tietokanta on Delphissä mahdollista toteuttaa monella eri tavalla, paikallisena tietokantana tai asiakas/palvelintietokantana. Yksinkertaisin näistä on paikallinen, joka sijaitsee kokonaan yhdessä koneessa. Asiakas/palvelin malli on toteutettu siten että tietokanta on palvelimella ja tähän on pääsy useammalla käyttäjällä. Tietokantoja kutsutaan usein joko yksi – tai kaksitasoisiksi tietokannoiksi. Yksitasoisessa tietokannassa muutokset tapahtuvat välittömästi. Kaksitasoisissa tietokannoissa ohjelmisto on yhteydessä tietokantaa jonkin tietokanta-ajurin avulla. [8, s. 593 – 594]

Delphissä tietokannan taulu koostuu vähintään kahdesta osasta: kentästä ja tietueesta. Kentät muodostavat taulussa olevan informaation otsikot, aivan kuten relaatiotietokannassakin on taulu ja sillä sarakkeet. Taulu sisältää myös tietueita jotka muodostuvat ryhmästä kenttiä. Aivan kuten edellisessä, on Delphin tietue sama kuin relaatiotietokannan rivi. Tällaisen yhden taulun tarkasteleminen ei yleensä riitä, vaan tarvitaan useiden taulujen yhdistelmiä. Tällöin puhutaan tietojoukoista (Data Set) joka saadaan tietokannasta esimerkiksi SQL-haun tuloksena. Tietojoukko voi sisältää informaatiota useasta eri lähteestä, se voi sisältää tietoa asiakastiedoista, tuotteista tai tilauksista. Vaikka informaatio tulee monesta erillisestä tietolähteestä, se kuitenkin esitetään yhtenä tietojoukkona. [8, s. 591 – 592.]

Delphin VCL-tietokantakomponentteja on kahdenlaisia: ei-visuaalisia sekä visuaalisia. Ei-visuaaliset datayhteyskomponentit (Data Access Component) vastaavat informaation välittämisestä tietokannasta visuaalisille komponenteille. Visuaaliset datatietoiset komponentit (Data Aware Components) vastaavat informaation välittämisestä käyttäjän sekä datayhteyskomponenttien välillä. Delphissä datayhteys- ja datatietoisien komponenttien yhdistämiseen käytetään DataSource-komponenttia (tietolähde). Tämän komponentin tarkoitus on tehdä tietokantakomponenttien yhteen liittämisestä helppoa. Kuvan 11 kaaviossa on esitetty VCL-tietokantakomponenttien arkkitehtuuri. Tästä huomataan että kaikki toiminnallisuus datayhteyskomponentin ja datatietoisien komponenttien välillä kulkee TDataSource-komponentin kautta. [8, s. 597; 7. s. 615.]



KUVA 11. Delphi VCL-tietokantakomponenttien arkkitehtuuri [8. s. 597]

Norellolla on käytössään ExpectIT:n valmistama INAP 2.0 tietokantakomponentti (Informix Native Access Pack). Tämä komponentti on datayhteyskomponentti joka tarjoaa rajapinnan Informix-tietokantapalvelimen ja datatietoisten komponenttien välille. Komponentti toimii aivan kuten edellä mainitut datayhteyskomponentit ja on tällä tavalla suoraan yhteensopiva Delphin muiden komponenttien kanssa.

5.5 Käyttöliittymän tietokantakomponentit

Delphissä graafiset tietokantakomponentit yhdistetään tietolähteeseen DataSource-ominaisuuden avulla. Tämä on yhteinen kaikilla tietokantakomponenteilla ja tekee tietokantasovellusten kehittämisestä tällöin nopeaa. Tämän lisäksi useimmilla tietokantakomponenteilla on DataField-ominaisuus jota käytetään liittämään komponentti johonkin tiettyyn tietokannan kenttään. Joillakin komponenteilla voi olla useita DataField-ominaisuuksia, esimerkiksi TDBGrid-komponentilla. [8, s. 643.]

Yleisimmin käytettyjä komponentteja kuitenkin ovat TDBGrid, TDBEdit ja DBNavigator. Näistä TDBGrid-komponentti on tarkoitettu taulukkomaiseen informaation esi-

tykseen ja on todella käytännöllinen koska ei tarvitse paljoa tilaa, suhteessa esitettyyn informaation määrään. TDBEdit-komponenttia voidaan käyttää yksittäisten tietueiden esittämiseen, esimerkiksi tuotekoodin esitykseen. TDBNavigator on komponentti jolla kontrolloidaan tietokannan tietueiden esitystä. Tämä tapahtuu komponentin sisältämien painikkeiden avulla. Näillä painikkeilla suoritetaan tietueissa liikkuminen sekä informaationkäsittely. [8, s. 644 – 645.]

6 TUOTANNONOHJAUS JA -SUUNNITTELU

Nykyisessä yritysten tuotannon toimintaympäristössä on käytössä verkostoajattelu. Ei riitä että ohjausjärjestelmät ajatellaan erillisinä osina yrityksen rakennetta vaan tätä tulisi ajatella koko yrityksen toimintaa tukevana systeeminä. Tällöin on tärkeitä kyetä takaamaan yrityksen eri osa-alueiden toimivuus kiinteässä yhteistyössä. [9, s. 23.]

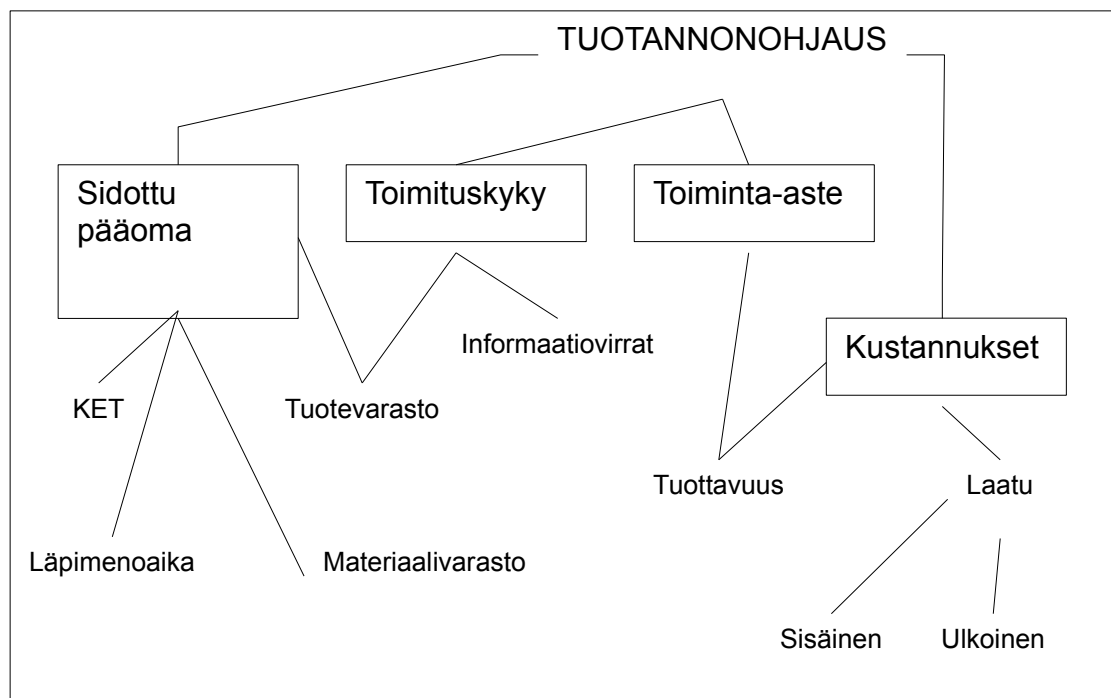
Materiaalin kulku talousjärjestelmässä voidaan jakaa kolmeen eri vaiheeseen: tuotanto, markkinointi ja kulutus. Nämä kolme vaihetta ovat sidoksissa toisiinsa. Tuotteen kulutus ei välttämättä tarkoita tuotteen joutumista kaatopaikalle, vaan tuote voi olla osana suurempaa kokonaisuutta, jolloin se kulutetaan uudessa tuotantoprosessissa. Tuotteen kilpailukykyä pyritään ylläpitämään tuotannonsuunnittelulla, tuotekehityksellä sekä tuotannonohjauksella. Toiminnan ohjauksen tarkoituksena on tuotannon ylläpito sekä tehokkuuden ja taloudellisuuden kohottaminen. [10, s. 7 – 9].

6.1 Tuotannonohjaus

Tuotannonohjauksella tarkoitetaan yleensä tuotantojärjestelmän eri osien yhteen sovitamista tuotantotavoitteiden mukaisesti. Tavoitteet ovat usein lähtöisin asiakkaiden tarpeista. Tuotannon ohjattavuudella tarkoitetaan järjestelmän kykyä saavuttaa sille asetetut operatiiviset ohjaustavoitteet. Mitä paremmin tuotantoprosessi ja tähän liittyvät resurssit voidaan ohjata uudelleen, sitä parempi ohjattavuus järjestelmällä on. [9, s. 23 – 24.]

Tietotekniikan merkitys tuotannonohjauksessa on tärkeätä. Tämä mahdollistaa nopean tiedonmuokkauksen informaatioksi sekä tämän ohjaamisen tuotantopisteisiin. Tuotantoprosessin aikana tietotekniikkaa voidaan käyttää hoitamaan sekä informaatio- että päätöksentekoon kuuluvia tehtäviä. Johdolle on mahdollista tuottaa seurantatietoa, josta nähdään yrityksen tavoitteiden toteutumista tai tästä poikkeamista. ATK-järjestelmä mahdollistaa tiedon välittämisen yksiköiltä toiselle jolloin toimintaa häiritsevät kyselyt jäävät pois. [10, s. 17 – 18.]

Tuotannonohjauksella on tarkoitus ohjata yrityksen tuotantojärjestelmää. Tämän tarkoituksena on saavuttaa yrityksen päämäärälle asetetut tavoitteet, siltä osin kun ne ovat tuotantojärjestelmästä riippuvaisia. Kuvassa 12 on esitetty tuotannonohjauksen rakenne. [9, s.24.]



KUVA 12. Tuotannonohjauksen rakenne [9. s. 25]

Toimitusaika kuvaa kuinka pitkään yritykseltä menee valmiin tuotteen toimittamiseen asiakkaalle. Tämä määräytyy koko järjestelmän toimitusketjun läpimenoajan perusteella. Asiakkaan kannalta tämä kuitenkin tarkoittaa sitä aikaa, minkä tuotteen tuottaminen kestää siitä lähtien kun asiakas tilaa tuotteen sekä saa valmiin. Tällöin on tärkeää varmistaa tilauksen nopea saattaminen tuotantoon. Tämä tarkoittaa toimintaa oh-

jaavien informaatiovirtojen nopeaa käsittelyä, jotta toimitusajoissa ei olisi suuria eroja. [9, s. 25.]

Läpimenoaika on yksinkertainen menetelmä arvioida tuotannon tehokkuutta. Tästä nähdään tuotantoon vaikuttavien tekijöiden kuten koneiden asetusajojen, toimitusajojen sekä huonojen aikataulujen vaikutus järjestelmään. [9, s. 25.] Yrityksen kannalta läpimenoajan tulisi olla mahdollisimman pieni. Tällöin tuotteet voidaan valmistaa resursseja hukkaamatta, eli mahdollisimman pienin kustannuksin. [10, s. 23].

6.1.1 Tuotantolajit

Tuotantolajit voidaan luokitella kolmeen eri osaan: varasto- ja yhtenäistuo-
tointo, sarjatuotanto sekä projektituotanto. Käytännössä kuitenkin tuotanto sisältää osia näistä kaikista tuotantomuodoista. Eri tuotantoluokkiin jakaminen on kuitenkin järkevää, sillä se selventää ohjausjärjestelmien vaatimuksia. [10, s. 12.]

Varastotuotannolla tarkoitetaan tuotantoprosessia josta puuttuu valmistusvaihe. Tämän tuotantolajin ohjauksen painopisteet keskittyvät tuotevalikoiman, varastoinnin ja tuotteiden logistiikan tehokkuuteen. Yhtenäistuo-
tointo tarkoittaa tuotantoa jossa valmistetaan vain yhtä tuotetta. Tällaisen tuotannon materiaalin ohjaus on yleensä pitkälle automatisoitu. [10, s. 12.]

Sarjatuotannossa pyritään tuottamaan sellaisia tuotesarjoja jotka muodostuvat samasta tuotteesta. Erikoistilauksesta valmistettujen tuotteiden valmistuksesta käytetään termiä yksittäistuo-
tointo. Projektituotannossa tuotetaan tuotteita joiden lähtökohtana on toteuttaa tuotteita joiden tuotanto ovat kertaluontoista ja tähtää ainutkertaiseen tavoitteeseen.[10, s. 12.]

6.1.2 Tuotannon tavoitteet

Tuotannossa toimitusaika- ja varmuus tarkoittavat sitä miten nopeasti voidaan tuottaa tuotteita sekä miten varmasti tuote tuotetaan määräaikaan mennessä. Yleensä kiinnite-

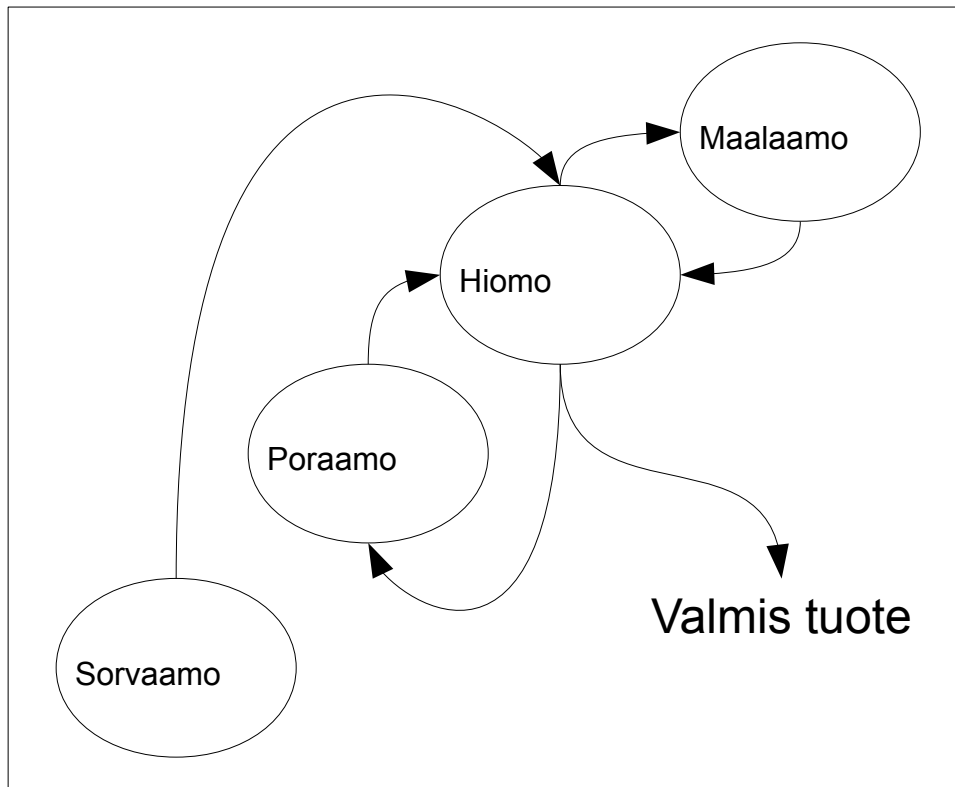
tään huomiota enemmän tuotteen tuotantonopeuteen kuin toimitusvarmuuteen. Useimmin kuitenkin tärkeämpää tuotannon kannalta on se kuinka varmasti tuote voidaan toimittaa luvatussa ajassa. Tuotannon joustavuudella tarkoitetaan sitä kuinka nopeasti yritys voi reagoida toimintaympäristön muutoksiin. Tämä on tärkeätä etenkin asiakaslähtöisissä yrityksissä. [9, s. 12 – 13.]

Laatu voidaan jakaa kahteen osaan: sisäinen ja ulkoinen laatu. Sisäisellä laadulla tarkoitetaan tuotannon valmistusprosessin laatua. Tämä tulisi aina maksimoida, sillä tästä ilmenee virheelliset osat mikä tarkoittaa materiaalin, työn ja kapasiteetin tuhlausta. Ulkoisella laadulla tarkoitetaan tuotteen ja siihen liittyvän palvelun kykyä tyydyttää asiakkaan tarpeet. Kustannukset tarkoittavat tuotteen valmistukseen kuluvia resursseja. [9, s. 14.]

6.1.3 Valmistusjärjestelmä

Valmistusjärjestelmäksi (layout) kutsutaan yrityksen koneiden sekä työnkulun muodostamaa ketjua. Tämä vastaa siitä miten työkoneet, työntekijät sekä materiaalivirtaus on toteutettu tuotannossa. Tämän päämääränä on myös saattaa sellaiset osastot yhteen joiden välillä on paljon toimintaa. Tällöin työntekijöiden ja materiaalien siirtäminen kohteesta toiseen ei kuluta resursseja paljoa. [11, s. 375.] Esimerkiksi keskusten koonpano on järkevää asettaa lähelle metalliosastoa, tällöin valmiiden metalliosien siirtämiseen keskusten tuotantolinjoille on mahdollisimman lyhyt.

Kuvassa 13 on esitetty toiminnallinen, eli funktionaalinen valmistusjärjestelmä. Tässä työ etenee ryhmästä tai osastosta toiseen työvaiheiden mukaisessa järjestyksessä. Tämän erityispiirteisiin kuuluu henkilöstön pitkälle erikoistuminen joka mahdollistaa teknisesti kehittyneen valmistuksen.



KUVA 13. Toiminnallinen valmistusjärjestelmä [9, s. 32]

Joustava valmistusjärjestelmä FMS(Flexible Manufacturing System) on valmistusjärjestelmä jossa tuotanto on pitkälle automatisoitua. Tällöin henkilöstölle jää vähän tehtäviä, käytännössä nämä ovatkin vain tarkastus- ja valvontatehtäviä sekä automaattisten robottien ohjelmointia. Tällaisen tuotantojärjestelmän tunnusomainen piirre on että se kykenee ylläpitämään miltei keskeytymätöntä tuotantoa. Esimerkiksi yöllä voidaan valmistaa seuraavana päivänä tarvittavia komponentteja. [9, s. 34 – 35.]

6.2 Tuotannonsuunnittelu

Tuotannonsuunnittelun lähtökohtana on tarjota välineet yritykselle joustavaan markkinoilla toimimiseen. Tämä tapahtuu sovittamalla markkinoiden tarpeet sekä tuotannon mahdollisuudet yhteen. Tällä pyritään takaamaan tuotantokapasiteetin kuormituksen tasainen käyttö sekä toimitusaikojen noudattaminen. Tuotannonsuunnittelun edellytyksenä on tulevan kysynnän sekä tiedossa olevan kapasiteetin tunteminen. [9, s. 37.]

Tuotannon työsuunnittelun tehtävänä on tilausten purkaminen tuotannon resurssien mukaisesti. Tähän sisältyy tilaukseen toteuttamiseen kuuluvien resurssien riittävyyden määrittäminen. Kun tarvittavat resurssit on määritetty, voi tuotannonsuunnittelija käynnistää tuotteen tuotannon. Tuotantosarjojen edetessä valmistuksessa, työsuunnittelun on valvottava töiden aikataulutusta. Mikäli valmistus poikkeaa aikataulusta on suunnittelun informoitava tuotantoa mahdollisista syistä, jotka aiheuttavat poikkeamia tuotteen läpimenoon. [10, s. 41.]

7 CASE: NORELCO OY

Norelco on suomalainen perheyritys joka suunnittelee ja valmistaa sähköjakelukeskuksia. Asiakaskunta muodostuu teollisuudesta, energiasektorista, liike- ja toimitiloista sekä asumisesta. Tuotteiden arvostus luodaan laajalla palvelulla, tinkimättömällä laadulla sekä jatkuvalla kehitystyöllä. Yritys sai alkunsa vuonna 1962 kun E.O. Härmäläinen aloitti elektroniikka- ja sähköalan liiketoiminnan Savonlinnassa. Nykyisin Norelcolla on toimipaikat suomessa Vantaalla, Kuopiossa ja Savonlinnassa sekä henkilöstöä 160.

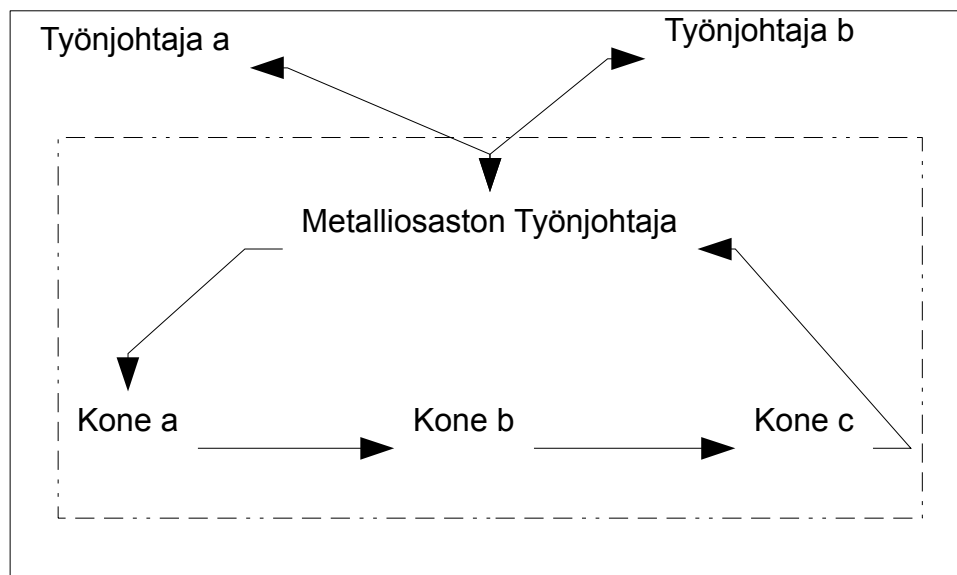
Norelco pyrkii mahdollisimman pitkälle vietyyn omavaraisuuteen tuotteiden tuotannossa. Tämä tarkoittaa Norelcon tuotteissaan käyttämien metalliosien omavaraista valmistusta raaka-aineesta alkaen. Tämä tapahtuu Norelcon metalliosastolla jolla valmistetaan suurin osa Norelcon käyttämistä metalliosista.

Norelcon metalliosasto valmistaa metalliosia komponentteina muiden osastojen tuotteiden tarpeisiin. Muiden osastojen metalliosien tarpeet lähtevät tilanteesta, kun asiakas tekee tilauksen kokonaisesta tuotteesta, esimerkiksi pienjännitekojeisto. Norelcon suunnittelussa suunnitellaan ensin pienjännitekojeisto sekä piirretään tuotteen rakenne. Tällöin myös voidaan joko käyttää jo olemassa olevia metalliosia tai suunnitella puuttuvat osat. Kun suunnittelu sekä piirtäminen on suoritettu, tilataan komponentit järjestelmään. Tämä voi sisältää myös muita komponentteja kuin metalliosia. Kun metalliosat ja muut tuotteeseen kuuluvat ovat valmiina, kootaan metalliosista sekä muista tuotteeseen kuuluvista komponenteista valmis tuote.

Tämän työn tarkoituksena on luoda Norelcon metalliosastolle uusi ATK-järjestelmä, joka integroidaan Norelcon jo olemassa oleviin toimisto-ohjelmiin. Tällöin on mahdollista toteuttaa saumaton tuotannon eteneminen, keskusten markkinoinnista ja tilauksesta, metalliosien tuotantoon saakka. Nimeksi Norelco on uudelle ohjelmistolle antanut NMetal(Norelcon Metalliosaston Tuotannonsuunnittelu- ja ohjausohjelmisto).

7.1 Norelcon metalliosasto

Tässä luvussa syvennyttään Norelcon metalliosaston toimintaan, jotta voidaan kartoittaa paremmin metalliosaston ATK-tarpeet. Tarkastellaan mitä erityispiirteitä on sekä kuvataan metalliosaston toiminta. Kuvassa 14 on kuvattu metalliosaston toiminnan pelkistetty kaavio. Tästä voidaan huomata metalliosaston sisäinen rakenne ja ulkopuolelle siirrettävät tapahtumat. Metalliosasto rakentuu työnjohtajasta sekä työkoneilla työskentelevistä metallialan ammattilaisista.



KUVA 14. Metalliosaston tuotannon pelkistetty kaavio

Metalliosaston tuotteet valmistetaan muiden osastojen tilausten pohjalta. Tämä tarkoittaa tilannetta jossa muiden osastojen työnjohtajat tilaavat metalliosastolta metalliosia, metalliosasto tuottaa osat ja tarjoaa muille osastoille valmiita metalliosia. Metalliosaston tuotanto alkaa siis tarpeesta metalliosalle, jolloin metalliosastolta tilataan metal-

liosia. On myös mahdollista että metalliosia on itsessään tuote, esimerkiksi sähkökeskuksen varasosa ovi. Kun metalliosia on tilattu, tulee metalliosaston tuotannonjohtajan hyväksyä tilaus. Tämän jälkeen tilauksen tuote-erän kuvaus siirtyy ensimmäiselle työkonelle.

Luvussa 6.1.3 kuvatut toiminnallinen sekä FMS-järjestelmä kuvaa hyvin Norelcon metalliosaston toimintaa. Norelcon metalliosasto muodostuu työkoneista joilla työntekijät valmistavat työvaihekohtaista tuotantoa. Tämän lisäksi käytössä on robotteja jotka mahdollistavat tuotteiden työvaiheiden automatisoitua tuotantoa. Jolloin voidaan ylläpitää keskeytymätöntä tuotantoa suurimman osan ajasta, jolloin jäljelle jäävä aika kuluu laitteiden asettamiseen.

Tuotteita valmistetaan sarjamuodossa, joka tapahtuu kierrättäen valmistettava metalliosa, osan valmistukseen käytettävillä työkoneilla. Valmistuksessa pyritään valmistamaan mahdollisimman suuria valmistuseriä yhdellä kertaa. Tarkoittaen että kierrätetään mahdollisimman paljon työvaiheitaan samankaltaisia tuotteita samaan aikaan. Tällä pyritään minimoimaan työkoneiden asetusten asettamiseen kuluva aika. Tämä ei aina kuitenkaan toimi, sillä voi olla tilauksia joilla syytä tavallista nopeampaan valmistumiseen. Tällöin tuote-erä on kiireellinen ja se tulisi valmistaa mahdollisimman nopeasti.

Kun tuotteesta saadaan työvaihe valmiiksi, siirretään se seuraavan vaiheeseen, kunnes kaikki tuotteen valmistukseen kuuluvat työvaiheet ovat käyty läpi ja tuote on valmis. Tällöin metalliosaston työnjohtaja kuittaa valmiiksi saadun työn, sekä ilmoittaa muiden osastojen työnjohtajille, valmiista metalliosista.

7.2 Metalliosaston tarpeiden kartoittaminen

Norelcolla oli tarve uudistaa metalliosaston ATK-järjestelmät. Tämän työn toteutuksessa ei tarkasteltu vanhaa järjestelmää tai sen toimintaa. Tarkoituksena oli toteuttaa aivan uusi järjestelmä uusine käyttötapoineen, jolloin vanhan järjestelmän syvällisemmästä analysoinnista ei olisi ollut hyötyä. Metalliosaston henkilökunnalla oli pyyntö

säilyttää uudessa järjestelmässä vanhan järjestelmän käyttöliittymän ulkoasu. Tällöin uuden järjestelmän käyttöönotto ei olisi vaikeaa.

Tarpeena olisi toteuttaa järjestelmä jolla voidaan hallita tuotteiden ylläpito, raportointi sekä tilaus. Tuotteiden ylläpitoon kuuluu tuotteiden sekä näiden attribuuttien luonti ja päivitys. Attribuuteilla tarkoitetaan osien ominaisuuksia, esimerkiksi väri tai varastopaikka.

Järjestelmän tarkoituksena on myös tarjota muiden osastojen työnjohtajille mahdollisuus tilata metalliostolta metallikomponentteja. Tämä tapahtuu metalliosien tilauksen kautta. Raportoinnilla halutaan seurata tietokannassa olevien metalliosien ajantasaista rakennetta sekä tilattujen metalliosien toimitustilaa. Tämän on myös tarjottava mahdollisuus kiireellisten metalliosien tilaukseen. Tällöin työnjohtajat voivat tilata kiireelliseen projektiin metallituotteita oikeassa ajassa. Jolloin projekti pysyy aikataulussa.

Suunnitteluosastolla tulee olla mahdollisuus ajantasaiseen metalliosien kuvausten päivitykseen, siten että metalliosastolla nähdään ajantasainen kuvaus metalliosien sen hetkistä spesifikaatioista sekä metalliosastolla tulisi olla mahdollisuus huomioida kun metalliosan kuvaus muuttuu. Suunnittelun jo olemassa olevalla järjestelmällä tulisi pystyä tilamaan metalliosia suoraan metalliosastolta, tai siten että tilaus siirretään projektia hoitavan osaston työjohtajalle.

Metalliosaston työnjohtajalla tulee olla mahdollisuus hyväksyä tai hylätä metalliosien tilaus. Tämän jälkeen hänen tulee pystyä kontrolloimaan metalliosien tuotantojonoja sekä saamaan ajantasaisia raportteja tuotannosta. Tuotantolinjoilla on työkonekohtaiset PC-tietokoneet, joissa ajetaan NMetal-ohjelmistoa. Näillä koneilla tulee olla mahdollista nähdä tuotannonsuunnittelijan määräämä tuotanto- konekohtainen tuotantojono. Työntekijöillä tulee olla myös mahdollisuus kirjata oma työnsä tulos järjestelmään ja siirtää tällä tavalla valmistuneet tuotteet, tuotteen seuraavaan työvaiheeseen.

8 NMetal-OHJELMISTO

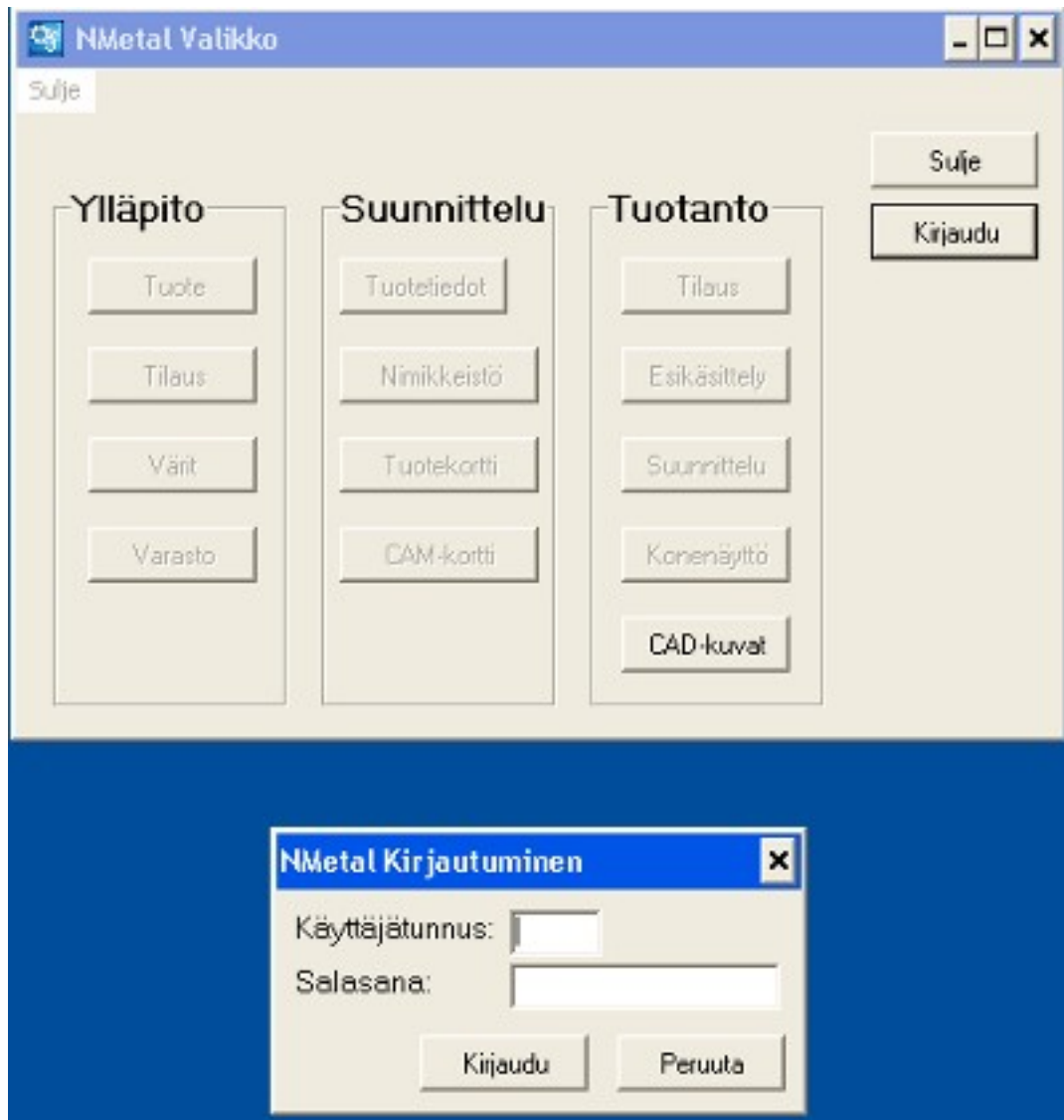
Tässä luvussa tutustutaan NMetal- ohjelmiston toteutukseen ja tähän vaikuttaviin päätöksiin. Rakenne on toteutettu siten, että ensin tutkitaan NMetal-ohjelmiston rakennetta, tämän jälkeen tarkastellaan kuinka metalliosat luodaan, kuinka tuotannon eri vaiheet on toteutettu sekä minkälaisia raportteja ohjelmalla on mahdollista toteuttaa.

8.1 NMetal-ohjelmiston rakenne

NMetal-ohjelmisto rakentuu neljästä pääkomponentista, ylläpito, suunnittelu, tuotanto sekä raportointi. Ohjelmistoon on toteutettu käyttäjätasojen perusteella toimiva autentikointijärjestelmä. Tämä tarkoittaa käyttäjätunnusten käyttöä sekä näiden tarjoamia erilaisia valtuuksia ohjelmiston käyttöön. Tällä voidaan toteuttaa tilanne jossa metallikoneen työstäjä ei voi tilata metalliosia, muuttaa näiden ominaisuuksia tai muokata toisten työntekijöiden työjonoja. Tästä huolimatta hänen tulee silti pystyä kirjata omien töidensä tulokset järjestelmään.

Ohjelmiston ulkoasussa on pyritty selkeyteen sekä nopeaan ohjelmiston oppimis- sekä käyttöönottonopeuteen. Tällä tavalla ohjelmiston käyttöönotto sekä uusien käyttäjien kouluttaminen on mahdollisimman nopeaa ja selkeää. Käyttäjät voivat nopeasti aloittaa työn eikä ohjelmisto aiheuta sen monimutkaisuudella haastetta työn etenemiseen. Tämä on pyritty toteuttamaan tapauskohtaisesti käyttämällä tilanteen sallimaan fonttikokoa sekä piilottamaan sellaista tietoa jota käyttäjä ei tarvitse. Kuten Metsämäki ilmaisee asian "Käyttöliittymä ei saa hukuttaa käyttäjää turhaan tietoon" [4, s. 59].

Käyttäjän tulee aina ensin kirjautua järjestelmään ennen ohjelmiston käyttöä. Kun käyttäjä lopettaa ohjelmiston käytön, tulee käyttäjän joko kirjautua ulos järjestelmästä tai sulkea ohjelma. Ohjelman sulkeminen kirjaa käyttäjän automaattisesti pois järjestelmästä. Kuvassa 15 on esitetty NMetal-ohjelman päävalikko sekä sisäänkirjaus. Päävalikko on suunniteltu siten että se on helppo käyttää ja kaikki oleelliset toiminnot ovat mahdollisimman lähellä ulkokerrosta.



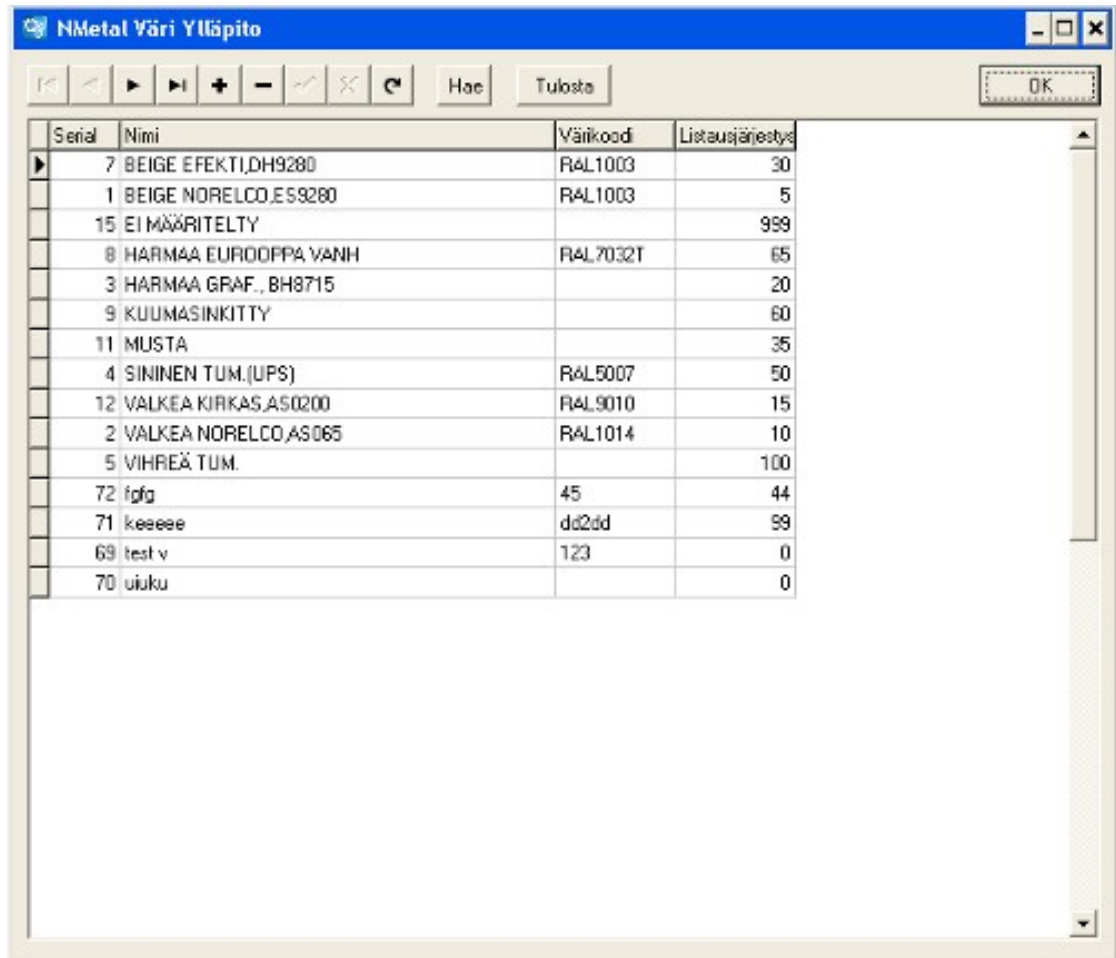
KUVA 15. NMetal-ohjelman sisäänkirjautuminen sekä päävalikko

Ohjelmiston päävalikon kautta voidaan käyttää kaikkia keskeisimpiä ohjelmistoon liittyviä ominaisuuksia. Sellaiset ominaisuudet joita käyttäjän käyttötaso ei salli hänen käytettävän, jätetään harmaiksi eikä käyttäjä voi näitä ominaisuuksia käyttää. Tällä tavalla voidaan myös estää esimerkiksi työkoneen käyttäjän muutosyritykset tuotetietoihin. Tämä toimii myös selkeyttäjänä, koska ominaisuuden ollessa pois käytöstä, ei tule ongelmaa jossa käyttäjä eksyisi ohjelmistossa sellaisiin ominaisuuksiin joihin häntä ei ole opastettu.

8.1.1 Metalliosien luominen

Suunnittelun tarkoituksena on tarjota puitteet metalliosien ylläpitämiseen sekä uusien metalliosien kuvausten luontiin. Metalliosalle on olemassa joukko pääominaisuuksia, kuten POL-koodi, nimike tai tuoteperhe. Tämän lisäksi tuotteeseen kuuluvat myös sen raaka-aine ja työvaiheet. Metalliosien työvaiheet muodostavat metalliosan valmistuksesta kuvauksen, missä järjestyksessä ja millä koneilla tuote tuotannossa valmistetaan. Työvaiheet määritellään näiden suoritusjärjestyksessä. Kaikkia työvaiheita ei välttämättä tarvitse toteuttaa, vaan tilauksen yhteydessä voidaan valita millä tuotantolinjalla tuote valmistetaan, eli mitä työvaiheita valmistukseen käytetään.

Sillä ohjelmiston koska käytössä on relaatiotietokanta, voidaan metalliosille asettaa helposti attribuutteja kuten väri. Tämä on mahdollista koska relaatiotietokanta mahdollistaa eri taulujen yhdistämisen JOIN- komennon avulla [1, s. 50 – 52]. Koska Norlcolla on käytössään useita eri välimalleja, tulee tuotteen tilaajan voida valita näistä väreitä tarvitsemansa. Tästä syystä käytössä oleville väreille on tarve luoda kuvaus. Tämä tapahtuu kuvan 16 väri-editorissa. Tätä ikkunaa voivat käyttää vain ylläpitoon kuuluvat henkilöt. Tarkka käyttöoikeuksien rajaaminen on tässä tarpeen, sillä kaikki muutokset mitä tehdään väri-editorilla. Vaikuttavat järjestelmän kaikkiin tuotteisiin.



Serial	Nimi	Värikoodi	Listausjärjestys
7	BEIGE EFEKTI, DH9280	RAL1003	30
1	BEIGE NORELCO, ES9280	RAL1003	5
15	EI MÄÄRITELTY		999
8	HARMAA EUROOPPA VANH	RAL7032T	65
3	HARMAA GRAF., BH8715		20
9	KUUMASINKITTY		60
11	MUSTA		35
4	SININEN TUM.(UPS)	RAL5007	50
12	VALKEA KIRKAS, AS0200	RAL9010	15
2	VALKEA NORELCO, AS065	RAL1014	10
5	VIHREÄ TUM.		100
72	fgfg	45	44
71	keeeee	dd2dd	99
69	test v	123	0
70	uiuku		0

KUVA 16. NMetal-ohjelmiston värien luonti ja muokkaus

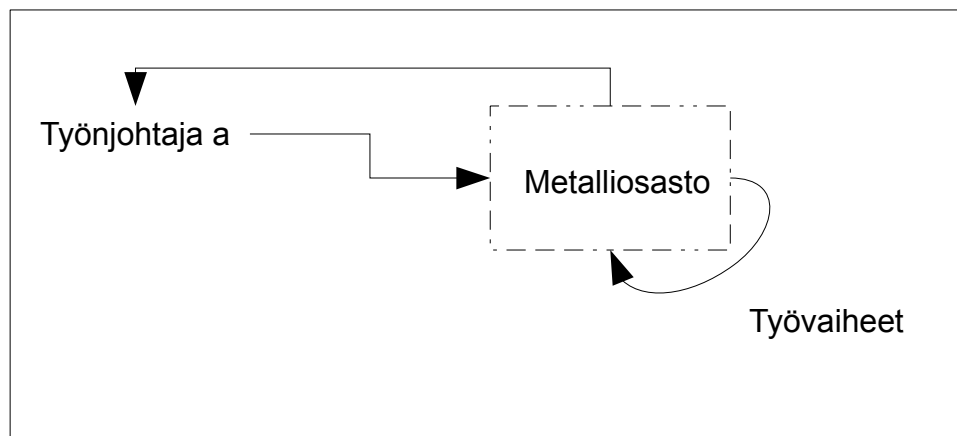
Kaikki värieditorissa määrätyt värit ovat käytössä muissa NMetal-ohjelmiston komponenteissa. Tällä tavalla saadaan keskitetty värien hallinta eivätkä uusien värien luominen tai vanhojen päivittäminen aiheuta suurempaa ongelmaa. Järjestelmässä on myös toisia samalla periaatteella toimivia tuotteiden attribuutteja.

Metalliosien muokkaaminen sekä uusien osien luominen tapahtuu NMetal-ohjelman tuote ikkunassa. Kuvassa 17 on esitetty tuotteiden luomiseen tarkoitettu ikkuna. Ikkuna muodostuu tuotteen tunnisteista kuten POL-koodi, tuotenimi, väri sekä joukko muita metalliosastolle tärkeitä elementtejä. Näiden lisäksi tuotteelle voidaan määrittää myös materiaalit, sekä tämän tuotteen valmistukseen tarvittavien materiaalien määrät.

Tuotannossa on myös tärkeätä pystyä jättämään viestejä tuotantotyöntekijöille. Tämä tapahtuu huomautus kentällä. Tämä on erityisen tärkeä kun tuotteen kuvausta on muokattu tai jos tuote on hieman perustuotteesta poikkeava. Huomiokentässä oleva viesti saatetaan tuotantotyöntekijöiden nähtäväksi ja ohjelmisto ilmoittaa tuotantotyöntekijää olemassa olevasta huomautuksesta. Tämä huomio näkyy myös tilaajalle, jotta tilaaja olisi myös tietoinen tuotetta koskevista erityismääreistä.

8.1.2 Tuotannon käsittely

Tuotanto muodostuu metalliosien tilauksesta, tilattujen tuotteiden käsittelystä sekä tilausten tuotannosta. Perusidea on kuvassa 18 esitetyn kaavion kaltainen. Tässä *työnjohtaja a* tilaa metalliosastolta tuotteen, metalliosasto ottaa tilauksen vastaan, asettaa sen tuotantoon ja kierrättää tuotetta metalliosastolla kunnes kaikki sen työvaiheet on saatu valmiiksi. Tämän jälkeen kuvan 18 *työnjohtaja a* näkee NMetal-ohjelmistolla tuotteen valmistuneen. Tällöin valmis tuote voidaan käydä hakemassa metalliosastolta.



KUVA 18. NMetal-ohjelmiston tilauksen pelkistetty kaavio

Metallituotteita tilatessa voidaan näiden ominaisuuksia muuttaa vapaasti. Esimerkiksi jos on tarve tilata vaalean tuotteen sijasta vaikkapa sininen, voidaan muutos tehdä helposti tilauksen yhteydessä. Tämä on kuitenkin rajoitettu siten, että värin kuvauksen on oltava valmiina tietokannassa. Tällä tavalla vältetään sellaisilta attribuuteilta, joille ei löydy kuvausta. Tilauksen jälkeen tilausta voidaan muokata vielä rajallisesti. Tilauksen muokkausta voidaan tehdä niin kauan, kunnes metalliosasto on ottanut tilauksen

tuotantoon. Tämän jälkeen tilausta voi enää muokata metalliosaton tuotannon suunnittelussa. Tilauksen yhteydessä järjestelmä laskee automaattisesti tilausmäärään tarvittavan oletusmääräisen raaka-aineen. Tämä helpottaa tuotantoa ja vähentää tilausten käsittelyyn käytettävää aikaa.

Kuvassa 19 on esitetty tuotteen tilausikkuna. Tällä voidaan tilata tuotteita sekä muokata buukkaamattomia tilauksia. Ulkoasun lähtökohtana oli tavoite saada käyttöliittymä toimimaan mahdollisimman nopeasti. Tämä tarkoittaa tilannetta, jossa tilauksen tekijällä olisi mahdollisuus muokata kaikkia tilauksen ominaisuuksia joihin hänen käyttäjätasonsa antaa mahdollisuuden. Tässä on pyritty tilanteeseen, jossa käyttäjän ei tarvitse käyttää hiirtä ensimmäisen tilauksen jälkeen, uutta tilausta tehdessään. Käyttäjä voi tällöin syöttää näppäimistöllä koneelle tietoa ja edetä automaattisesti kentästä toiseen.

NMetal Tilaus

Navigation: [Previous] [Next] [Home] [Cancel] [OK] [Hae] [Listaus] [E-tilaus]

Tilaus: 1/4 jonka Tila: Aloittamaton

Tilaaaja: 987 | Jaatinen

Tilaus pvm: 24.2.2011

Koodi: M90000004D | ESIMERKKI OSA 1

Tilaus määrä: 8 | Kp/vaihto: 0 | Kulutus/vko: 2

Toimitus pvm: 4.3.2011 | ☐ Kiire | Toimitus aika: 9

Pintokäsittely: 1 | Valm.nro: 0

Väri: 12 | VALKEA KIRKAS,AS0200 | RAL9010

Materioli: 01260132 | KYVA DC01 AMO 3X1000X2000

Varustopaikka: 1SA | FGs

Alihenkintä: | Pien-N-osa: ☐

Tuoteperhe: 10900 | VARAOSAT

Työväiheet: ☒ T

Huom(Tilaus):

Huom(Tuote): HEI TUOTANTO

KUVA 19. NMetal-ohjelmiston tilausikkuna

Tilausta tehdessään tilaaja näkee tuotannon huomautuksen ja voi myös kirjoittaa erilliseen tilausta koskevaan huomautuskenttään oman huomautuksensa. Tämä on tärkeää esimerkiksi erikoisia metalliosia tilattaessa tai kiiretilanteessa, jolloin on hyvä perus-

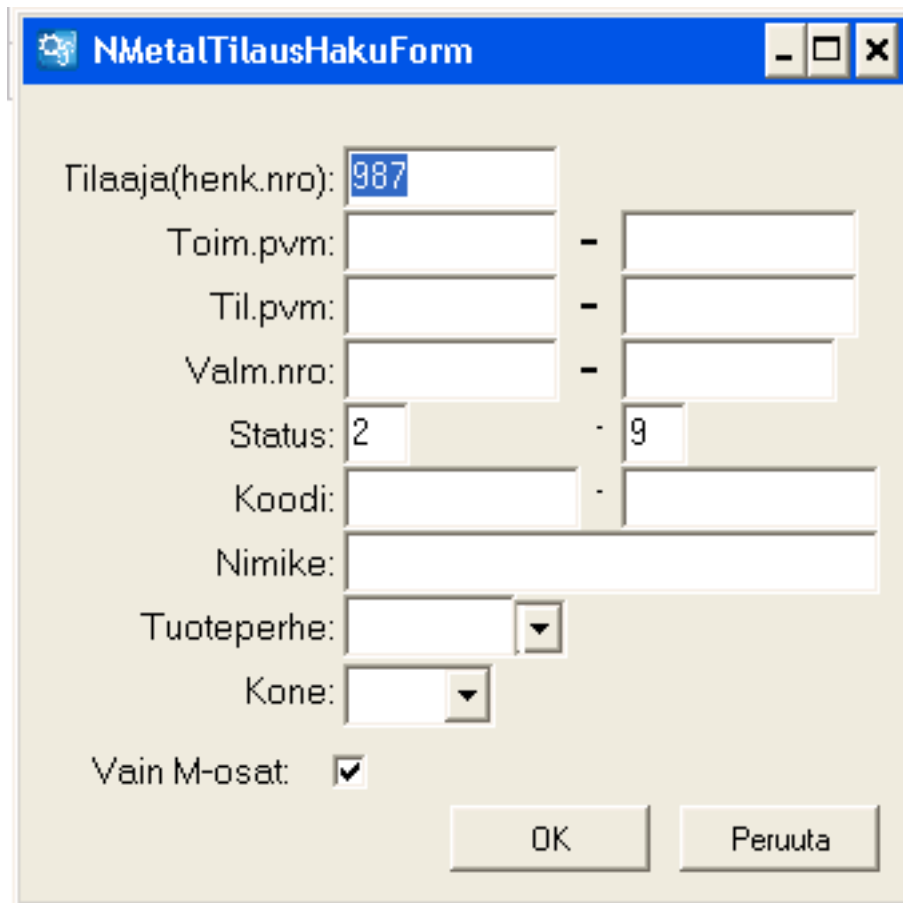
tella kiireen syytä. Tilaus on mahdollista määrittää kiireelliseksi, joka tapahtuu valitsemalla *kiire*-valinta. Tällä tarkoitetaan tilannetta jossa tilaaja haluaisi tilauksen valmistuvan mahdollisimman pian. Tällöin metalliosaston työnjohtaja voi järjestellä työjonot siten, että kiirelipulla merkityt työt valmistuvat muita töitä nopeammin.

Tilaajien tulee antaa kaikki tilausikkunassa olevat metalliosan tiedot. Tämä on varmistettu käyttöliittymätasolla siten, että tilauksen hyväksymishetkellä ohjelma ilmoittaa puuttuvista tiedoista. Tässä sovellettiin Metsämäen ajatusta vuorovaikutteisuudesta [4, s. 63]. Tällöin käyttäjä voi halutessaan joko perua tilauksen ja olla tilaamatta tuotetta, tai hän voi korjata puuttuvat tiedot ja tilata tuotteen. Tällä on pyritty tilanteeseen jossa tuotannonsuunnittelijan ei tarvitse turhaan selvittää usein puuttuvien tietojen tarkoitusta.

Tilausta tehdessä näytetään myös kaikki samaa tuotetta olevat sellaiset tilaukset, joiden tuotantoa ei vielä ole aloitettu. Tämä on tarkoitettu tilanteeseen, jossa järjestelmässä on jo olemassa jonottava tilaus kyseiselle tuotteelle. Tällöin tilaaja voi pyytää jo olemassa olevan tilauksen kokoa kasvatettavan. Tällä tavalla pyritään luomaan suurempia tilauseriä sekä voidaan antaa mahdollisuus suuren tilauserän kokoamiseen myös tilaajille. Kaikkien etu on, että tuotteita saadaan valmistettua mahdollisimman laadukkaasti ja nopeasti.

Järjestelmässä jo olevia tilauksia voidaan myös seurata. Kuvassa 20 on esitetty tilausten hakuun käytettävä ikkuna. Tästä huomataan, että ohjelmisto sisältää useita eri hakuvaihtoehtoja sekä näiden yhdistelmiä. Tällä tavoin voidaan rajata seurattava tilaus hyvinkin tarkasti, sekä hakea suuriakin joukkoja. Haussa on myös mahdollista rajata halutaanko tarkastella vain tilattuja tuotteita, vaiko myös tilausten yhteydessä tilattuja tuotteiden osia.

Hakujen yhdistäminen on mahdollista sillä relaatiotietokannat mahdollistavat erilaisien liitosten sekä predikaattilogiikkaan perustuvien lauseiden yhdistämisen. Tästä Hovi [1, s. 64 – 65] esittää keinon IN-alikyselyjen käyttöön. Tätä menetelmää käyttäen metalliosien tilaukset voidaan rajata näiden attribuuttien sekä metalliosan kuvauksen perusteella.



The screenshot shows a Windows-style application window titled "NMetalTilausHakuForm". The window contains several input fields and controls for searching orders:

- Tilaaaja(henk.nro):** A text box containing the value "987".
- Toim.pvm:** A text box for the start date, followed by a hyphen and another empty text box.
- Til.pvm:** A text box for the end date, followed by a hyphen and another empty text box.
- Valm.nro:** A text box for the production number, followed by a hyphen and another empty text box.
- Status:** A text box containing "2", followed by a hyphen and a text box containing "9".
- Koodi:** A text box for the code, followed by a hyphen and another empty text box.
- Nimike:** A large text box for the description.
- Tuoteperhe:** A text box followed by a dropdown arrow.
- Kone:** A text box followed by a dropdown arrow.
- Vain M-osat:** A checkbox that is currently checked.
- Buttons:** "OK" and "Peruuta" (Cancel) buttons at the bottom right.

KUVA 20. Tilausten hakuun käytettävä ikkuna

Tilausikkunan listaus-painikkeesta, voidaan helposti edetä tilausten seurantaan. Tilausten seurannan avulla tilaaja voi nopeasti tarkastaa tuotannossa olevien tuotteiden tilanteen. Tällä tavalla tilaajat voivat arvioida metalliosien valmistumisen ja suunnitella omaa tuotantoaan. Kuvassa 21 on esitetty metalliosien seurantaan tarkoitettu ikkuna. Tällä ikkunalla voidaan seurata tarvittaessa kaikkien tilausten etenemistä metalliosastolla. Tämän ikkunan sisällöstä voidaan myös tulostaa listaus paperille.

POL-Koodi	Nimike	Tilauksen tila	Vaiheet	%-osat	Materiaali	Tilmäärä	Toimipvm	Kite	PK	Vähä	Varastopäi
N92000001	ESIVERKKI OSA 2	Alustuksen	T17, E12, H12	0%	KW4 D-0014M0 3x100x62000	6	25.9.2011	0	1	HÄJEE	154
N92000001B	ESIVERKKI OSA I	Alustuksen	T17	0%	KW4 D-0014M0 3x100x62000	6	25.9.2011	0	1	HÄJEE	154
N92000001B	ESIVERKKI OSA I	Alustuksen	T17	0%	KW4 D-0014M0 3x100x62000	6	25.9.2011	0	1	HÄJEE	154
N92000001B	ESIVERKKI OSA I	Alustuksen	T17	0%	MUSU D=510+2275VSC	16	25.9.2011	1	1	MIETÄ	KAL

KUVA 21. NMetal-ohjelmiston tilausten seurantaikkuna.

Kun metallituote on tilattu, jää se odottamaan että metalliosaston työnjohtaja ottaa työn tuotantoon eli buukkaa työn. Ennen tilauksen buukkaamista voi tilaaja muokata tilausta vapaasti hänen käyttötasonsa sallimalla tavalla. Kun tuote on buukkaamaton voi metalliosaston työnjohtaja muuttaa tilatun tuotteen työvaiheita ja tähän sisältyviä metalliosia. Hän voi myös muokata tilauksen määrää tai poistaa tilauksen. Kun metalliosaston työnjohtaja buukkaa tilauksen, siirtyy tuote tuotantoon.

Kuvassa 22 on esitetty metalliosien tilausten vahvistukseen käytettävä ikkuna. Tässä ikkunassa voidaan tilaus hyväksyä, jättää roikkumaan tai poistaa se. Tästä ikkunasta voidaan myös hallita tilauksen tuotteisiin kuuluvia työvaiheita sekä määrittää tilauksen määrä uudelleen. Ohjelma myös näyttää jo tuotannossa samaa tuotetta olevat tilaukset. Tilausten vahvistukseen käytettävällä ikkunalla voidaan tarkastella sekä muokata tilauksen yhteydessä tilattuja metalliosaan kuuluvia raaka-ainekomponentteja, sekä näiden ominaisuuksia.

The screenshot shows a software window titled "Metalliosien tilaustyyppiä". It contains a table with columns: +/-, PDI-Koodi, Nimi, Ais, Pn, Epä/Alus, Til.määrä, Toimipen, Käre, Materiaali, PK, Väri, and N.osat. The table lists several orders for metal parts, including "ESMER900 OSA 2" and "ESMER900 OSA 1".

+/-	PDI-Koodi	Nimi	Ais	Pn	Epä/Alus	Til.määrä	Toimipen	Käre	Materiaali	PK	Väri	N.osat
-	PR0000004	ESMER900 OSA 2			0	12	25.3.2011	0	01250132	1	12	0/0
-	PR0000004	ESMER900 OSA 2			0	9	25.3.2011	0	01250132	1	12	0/0
-	PR0000004E	ESMER900 OSA 1			0	16	25.3.2011	0	01250132	1	12	0/0
-	PR0000004E	ESMER900 OSA 1			0	9	4.3.2011	0	01250132	1	12	0/0
-	PR0000004E	ESMER900 OSA 1			0	9	25.3.2011	0	01250132	1	12	0/0
-	PR0000004E	ESMER900 OSA 1			0	16	25.3.2011	1	01250501	1	11	0/0

KUVA 22. Metalliosien tilausten vahvistus

Kun tilaus on saatettu tuotantoon, työnjohtajan on nähtävä eri koneiden työjonot, näiden valmistumiseen kuluva aika sekä tuotteiden työvaiheet. Hänen on voitava asettaa tuotteiden työkonelkohtainen tuotantojärjestys haluamallaan tavalla. Tämä on tärkeätä esimerkiksi levyntyöstökoneilla joissa voidaan yhdellä raaka-aineen vaihdolla toteuttaa mahdollisimman monia samasta raaka-aineesta valmistettuja tuotteita. Tällöin raaka-aineen vaihtoon ei kulu ylimääräistä aikaa.

Tuotannonsuunnittelija voi muokata tuotannonsuunnitteluun käytettävällä ikkunalla tuotantotyövaiheeseen kuuluvien tuotteiden tuotantojärjestystä. Tällä ikkunalla voidaan nähdä tuotannossa olevat työvaiheet ja näiden työjonojen sisältämät tuotteiden määrät sekä työajat. Työjonojen suunnittelu on toteutettu siten että, tästä näkyvät tietyn työkoneen tuotantojonot.

Tällä on pyritty mahdollistamaan tuotantojonojen muokkaaminen siten, että tämä mahdollistaa tuotannonohjauksen sekä tarjoaa puitteet luvussa 6.1.2 kuvattuun tuotannon tavoitteiden saavuttamiseen. Tällöin voidaan saavuttaa joustavuutta sekä taloudellisuutta, sillä tuotteiden valmistukseen voidaan puuttua työvaihe kohtaisesti, joka mahdollistaa kiireellisten tuotteiden läpimenon ja aikataulussa pysymisen. Toisaalta on myös mahdollista puskuroida tuotantoa. Tämä tapahtuu keräämällä mahdollisim-

man paljon työvaiheeltaan samankaltaisia tuotteita yhteen, jolloin työkoneiden asetusajat ovat pieniä.

Tuotannonsuunnittelijan on myös mahdollista nähdä koko tuotetta koskeva informaatio. Tällöin hän voi esimerkiksi ottaa yhteyttä tilaajaan ja ilmoittaa tuotteen valmistuksessa tapahtuneisiin muutoksiin tai kysyä lisätietoja tilattuun tuotteeseen liittyen. Järjestelmään on myös lisätty järjestelyä helpottavia ominaisuuksia. Esimerkiksi tuotannonsuunnittelijan on mahdollista automaattisesti järjestellä tuotantokoneilla olevien tuotteiden tuotantojärjestystä. Tämä on mahdollista tehdä esimerkiksi tuoteryhmittäin, tuotteiden toimituspäivämäärän tai materiaalin perusteella. Tämä on tärkeää sillä maalaamossa halutaan maalata yhtäjaksoisesti mahdollisimman monta samanväristä tuotetta.

Tuotteiden ryhmittäminen on mahdollista sillä relaatiotietokanta mahdollistaa SQL-kielen kanssa tuloksen järjestämisen halutulla tavalla. Bowman ym. [2, s. 144 – 146] ovat esittäneet esimerkkejä ORDER BY -ehdon käytöstä. Tätä voidaan käyttää järjestelemään metalliosien työvaiheet esimerkiksi POL-koodin tai värin mukaan.

Tuotannonsuunnittelija voi muokata tuotantokone kohtaisesti työjonoa siten, että hän valitsee työjonoon lisättävät tuotteet. Lisätyt tuotteet näytetään tuotantokoneiden työjonoissa ensimmäisinä ja näiden jälkeen tulee muu jono. Mikäli mitään tuotetta ei ole valittu, näytetään tuotantojono sellaisenaan. Tällä tavalla varmistetaan myös tuotannon eteneminen vaikka tuotannonsuunnittelija puuttuisi.

Jokaisella tuotantokoneella on PC, josta tuotannontyöntekijä näkee metalliosan työvaihetta koskevan tiedon. Työntekijä myös kirjaa oman tuotantonsa järjestelmään tällä ikkunalla. Tuotantokoneilla tapahtuva tuotteiden valmistumisen kirjaaminen on pyritty tekemään mahdollisimman yksinkertaiseksi sekä nopeaksi toimenpiteeksi. Työvaihetta koskeva informaatio on pyritty esittämään mahdollisimman selkeästi. Tässä ikkunassa tuotteiden esiintymisjärjestys on tuotannonjohtajan määräämä. Tuotantotyöntekijä voi kuitenkin järjestää listaa tietyin rajauksin, esimerkiksi maalaamossa voidaan tuotannonjohtajan määräämä lista järjestää tuotteiden värien mukaan tai levytyökeskuksella materiaalin.

Kun työvaihe on saatu valmiiksi, tuotantotyöntekijän on mahdollista tulostaa tästä saattokortti. Tämä lisätään valmistetun tuote-erän mukaan ja saattokortin tarkoituksena on kertoa seuraavalle tuotantotyövaiheelle mistä tuote-erästä on kyse ja minne se on matkalla. Tällä tavoin pyritään ehkäisemään tuote-erien sekoittuminen eri tuotantovaiheissa. Mikäli tuotannossa tuote-erä jää kesken, voidaan se pilkkoa osiin ja tieto jo valmistuneesta määrästä sekä siitä kuka valmisti, jää järjestelmään. Tällöin seuraava työkoneen käyttäjä voi jatkaa siitä mihin edellinen työntekijä jäi.

8.1.3 Raportointi

NMetal-ohjelmisto tarjoaa laajan raportointialustan. Miltei kaikesta mitä NMetalilla tehdään, voidaan ottaa tulostuksena raportti. Keskeisimpiä raportteja ovat tuotetietokanta, tilaukset ja tuotannon seuraaminen. Tilausten seuraaminen tapahtuu miltei reaaliajassa. Tämä mahdollistaa nopean töiden etenemisen eri osastojen välillä.

Liitteenä 2 on esimerkkinä raportti jota käytetään tilausten seurantaan. Tästä raportista käy hyvin ilmi missä vaiheessa tuotteet ovat, sekä tilattujen tuotteiden ominaisuudet kuten tilausmäärä, tuotteen varastointipaikka tai toimituspäivämäärä. Koska tilattuja tuotteita on hyvin paljon, voidaan raporttia rajata esimerkiksi tiettyyn tilaajaan, tuotteeseen tai vaikkapa toimituspäivämäärään. Tätä raporttia voidaan käyttää tilanteessa, jossa haetaan metalliosastolta valmistunutta tuotetta. Tästä raportista näkyvät tuotteen varastointipaikka sekä tunnistamiseen tarvittavat tiedot.

Tuotannon seurantaan NMetal tarjoaa liitteenä 2 olevan raportin, jolla voidaan seurata tuotteiden etenemistä tuotantokoneilla. Tällä nähdään missä tilanteessa tuotteet ovat, paljonko tuotteita on tilattu sekä paljonko näitä on kesken. Tuotantotilanteista on myös mahdollista ottaa erinäisiä kuvauksia. Näistä keskeisimpiä ovat tuotantokoneiden työjonojen tarkastelut. Näistä nähdään tuotantokoneilta valmistuneiden tuotteiden määrät sekä keskeneräiset tuotantojonot. Tällä tiedolla voidaan ennustaa tulevien tilausten toimitusnopeutta ja suunnitella tuotantoa. Liitteenä 3 on esitetty raportti, jossa tarkastellaan keskeneräisiä tuotteita. Tästä nähdään tuotteiden tilaus- ja tuotantomäärät, sekä missä työvaiheissa tuotteet ovat. Tuotteiden työvaiheet näkemällä tuotannon-

suunnittelija voi helposti suunnitella tuotantojonoja siten, että mahdollisimman paljon tuotannosta valmistetaan yhtäjaksoisesti.

Metallituotteiden varastointiin käytetään lavakorttia. Tämän kortin tarkoituksena on toimia varastopaikan tunnuksena. Tämä kortti kiinnitetään varastopaikkaan ja metalliosaston työntekijät voivat tämän kortin avulla nähdä mitä varastopaikassa on. Tämän kortin toteutuksessa sovellettiin typografiaa. Lavakortin viivakoodi toteutettiin viivakoodi 3of9- määritelmän mukaisesti.

9 PÄÄTÄNTÖ

Opinnäytetyön tarkoituksena on luoda kuvaus Norelcolle tekemästani NMetal- ohjelmistosta. Valmistuneen ohjelmiston avulla on tarkoitus edesauttaa Norelcon metalliosaston toimintaa nopeaksi sekä luotettavaksi. NMetal-ohjelmisto on otettu käyttöön ja käyttäjien palaute on ollut hyvää. Käyttäjiltä tullut palaute on koskenut ohjelmiston yksinkertaisuutta, selväkielisyyttä sekä nopeaa toimintaa vastetta. Myös ohjelman käyttöönoton sujuvuutta on kiitetty.

Ohjelmistosta kuitenkin jäi puuttumaan vielä metalliosien yhdistäminen suuremmiksi tuotantokokonaisuuksiksi sekä CAD- kuvien esittäminen suoraan tuotantokonekohtaisella PC:llä. Etenkin metalliosien tilausten saattaminen suuremmiksi tuotantokokonaisuuksiksi on ollut ongelmallista.

Tähän mennessä tilanne on ratkaistu lisäämällä yhdistettävän tuotteen tilausmäärä toisen samankaltaisen tuotteen tilausmäärään ja poistamalla se tilaus, jonka tilausmäärä lisättiin. Tällöin tuotteiden tuotantomäärät ovat samat kuin tilattujen. Ongelmana edellä kuvatussa toimenpiteessä on tilauskohtainen seurannan menettäminen.

Tietokannan kuormittumien on eräs vaikeimmista ongelmista joita olen huomannut NMetalin toiminnassa. Tuotantotietokannassa on nyt kirjoitus hetkellä noin 7000 tuotetta tilattuna ja näiden työvaiheita on 16000. Tällaisen informaatiomäärän käsittely ja sen nopea saattaminen käyttäjille on ollut työn eräs vaativimmista haasteista. Tulevaisuudessa tähän ongelmaan voisi ehkä olla ratkaisuna siirtää tuotannonaikaisesta taulusta valmiit tuotteet toiseen samankaltaiseen tauluun, jolloin ne eivät hidastaisi tuotantojärjestelmää.

Delphi on kehitystyökaluna todella nopea ja sillä voi saada aikaan nopeasti paljon ohjelmistoa. Tästä syystä ohjelmisto suunnittelun teoria on keskeisessä asemassa. Mielestäni tämä tarkoittaa yleistä ohjelmointiin liittyvää ymmärtämistä. Koska teorian ja hyvien ohjelmointimallien puuttuminen voi aiheuttaa sellaisen ohjelmakoodin tuottamisen, jonka ylläpitäminen on myöhemmin vaikeata. Tästä syystä olisin toivonut koulutukseltani parempaa pohjaa itse ohjelmointimalleihin, en ohjelmointikieleen.

LÄHTEET

- 1 Hovi, Ari. SQL-ohjelmointi Pro Training. Jyväskylä: Satku-kauppakaari. 2000.
- 2 Bowman, Judith S. Emerson, Sandra L. Darnovsky Marcy. The practical SQL handbook : using SQL variants. Boston: Addison-Wesley. 2001.
- 3 Polvinen, Timo. Tietokannat käytännön työssä. Porvoo: WSOY. 1999.
- 4 Metsämäki, Markku. Graafinen käyttöliittymä. Helsinki: Painatuskeskus Oy. 1995.
- 5 Shneiderman, Ben. Designing the user interface: strategies for effective human-computer interaction. Reading: Addison-Wesley. 1998.
- 6 Järvinen, Jani. Piispa, Juha. Delphi sovellusten opas. Tummavuoren kirjapaino: Teknolit Oy. 2000.
- 7 Gamma, Erich. Helm, Richard. Johnson, Ralph. Vlissides, John. Olio-ohjelmointi suunnittelumallit. Helsinki: Oy Edita Ab. 2001.
- 8 Reisdorph, Kent. Delphi 5 Trainer Pro. Jyväskylä: Oy Edita Ab. 2000.
- 9 Miettinen, Pauli. Tuotannonohjaus- ja logistiikka. Helsinki: Painatuskeskus Oy. 1993.
- 10 Airas, Martti. Yrityksen tuotannonohjaus. Lappeenranta: Etelä-Saimaan Kustannus Oy. 1985.

Delphi 5- kehitysalustan tuottama koodi

```
unit Unit3;

interface

uses

    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

type
    TForm3 = class(TForm)
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form3: TForm3;

implementation

{$R *.DFM}

end.
```

LIITE 2.

NMetal-ohjelmiston tilausten seurantaan käytettävä raportti

NORELCO, Metalliosasto

Tilattu materiaali

17.3.2011
Sivu 1 / 1

POL-Koodi	Nimike & Materiaali	Valmno	Tila	Vaiheet	N-osat	Til.Määrä	Toim.pvm	Klire	PK	Väri	Varasto
M98000004	ESIMERKKI OSA 2 KYVA DC01AMD 3X1 000X2000	0	Buikkaamat	T:2, E:2, X:2	0/0	8	25.3.2011	0	1	12	ISA
M98000004B	ESIMERKKI OSA 1 KYVA DC01AMD 3X1 000X2000	0	Buikkaamat	T:2	0/0	8	4.3.2011	0	1	12	ISA
M98000004B	ESIMERKKI OSA 1 KYVA DC01AMD 3X1 000X2000	0	Buikkaamat	T:2	0/0	8	25.3.2011	0	1	12	ISA
M98000004B	ESIMERKKI OSA 1 KUSI DX51D+Z275MBC 1.0X1250X2500	0	Buikkaamat	T:2	0/0	16	25.3.2011	1	1	11	KAL

LIITE 3.**NMetal-ohjelmiston tuotannon seurantaan käytettävä raportti**

NORELCO		KESKENERÄISET TUOTTEET		AJALTA 18.3.2011 - 17.5.2011			SIVU 1 / 1	
		TULOSTETTU 18.3.2011 8:56:59		STATUS 2 - 8 ja KONEET A - Z				
KOODI	NIMITYS	TILATTU	KESKEN	VALMI	VAIHE	VALM.NRO		
M98000004	ESIMERKKI OSA 2	8	8	_____	X	0		
M98000004B	ESIMERKKI OSA 1	24	24	_____	T	0		